Modern Applied Science; Vol. 12, No. 1; 2018
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Supernova Optimizer: A Novel Natural Inspired Meta-Heuristic

Amjad A. Hudaib[1] & Hussam N. Fakhouri[1]

[1] King Abdullah II School of Information Technology, University of Jordan, Amman, Jordan

Correspondence: Amjad Hudaib[1], King Abdullah II School of Information Technology, University of Jordan, Amman, 11942, Jordan. E-mail: ahudaib@ju.edu.jo, h.fakhouri@ju.edu.jo

Received: November 2, 2017 Accepted: December 10, 2017 Online Published: December 23, 2017

doi:10.5539/mas.v12n1p32 URL: https://doi.org/10.5539/mas.v12n1p32

Abstract

Bio and natural phenomena inspired algorithms and meta-heuristics provide solutions to solve optimization and preliminary convergence problems. It significantly has wide effect that is integrated in many scientific fields. Thereby justifying the relevance development of many applications that relay on optimization algorithms, which allow finding the best solution in the shortest possible time. Therefore it is necessary to further consider and develop new swarm intelligence optimization algorithms. This paper proposes a novel optimization algorithm called supernova optimizer (SO) inspired by the supernova phenomena in nature. SO mimics this natural phenomena aiming to improve the three main features of optimization; exploration, exploitation, and local minima avoidance. The proposed meta-heuristic optimizer has been tested over 20 will known benchmarks functions, the results have been verified by a comparative study with the state of art optimization algorithms Grey Wolf Optimizer (GWO), A Sine Cosine Algorithm for solving optimization problems (SCA), Multi-Verse Optimizer (MVO), Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm (MFO), The Whale Optimization Algorithm (WOA), Polar Particle Swarm Optimizer (PLOARPSO) and with Particle Swarm Optimizer (PSO). The results showed that SO provided very competitive and effective results. It outperforms the best state-of-art algorithms that are compared to on the most of the tested benchmark functions.

Keywords: numerical optimization, meta-heuristic, function optimization, real time optimization

1. Introduction

The true beauty of algorithms inspired by nature is transforming the best solutions that could be created by nature to technology. Because of this fact, we have the ability to describe and solve complex problems with very simple initial conditions and rules that do not carry enough knowledge, about the nature of search space (Langdon, 2007) (Lugmayr et al., 2013). By carefully studying each natural phenomenon or complex interaction between organisms, from microorganisms to full human beings, it will let us see that nature always found the optimal strategy, ex: balancing the ecosystem, maintenance of diversity, adaptation, physical phenomena, such as the formation of rivers, the movement of clouds; movements of bird folks, The solution of these processes is simple, and the results are amazing. Problems in computer science have much in common with problems of nature; therefore, a mapping between nature and technology is possible calculations, inspired by nature, cover a wide range of applications, including computer networks, security, robotics, bio-medical equipment, control and parallel processing systems, data mining, energy systems, technology production and much more. (Lugmayr et al., 2013).

Classical methods for solving problems include two branches: mathematical and heuristics. Heuristic approach used in solving complex optimization tasks, especially where traditional methods fail. Heuristics, imitate the strategy of nature, it use a lot of random solutions, which is classified as a special class of randomized algorithms.

Meta-heuristic algorithms include (Zhang, 2015): a) choosing the correct representation of the problem; b) assessing the quality of the solution using the fitness function; c) defining operators in such a way as to obtain a new set solutions. The most prevalent and successful classes of this type are, evolutionary algorithms and algorithms based on the swarm, inspired natural evolution and based on the collective behavior of animals. The optimization method using a swarm of particles is traditionally used for solving the problem of searching for the global optimum of a multidimensional function, where it showed the very good results (Bonyadi, and Michalewicz, 2017)

Solving optimization problems can be performed by Swarm intelligence (SI) based algorithms, Bio-inspired algorithms, natural inspired, Physics and Chemistry inspired and Mathematical methods as shown in figure 1(Clerc, 2017)

In general, met-heuristic algorithms aim to find the extremum of the optimized problem, at which the value of the objective function is consistently improved until extremum point is found. Depending on whether the algorithm can be local or global, they are divided into algorithms of local and global search. Local extremum search algorithms are designed to determine one of the extremums on the set of admissible solutions in which the objective function the maximum or minimum value. To find the extremum when the form of the optimized function is not fully known, or its structure is too complex, the methods swarm intelligence are used. The efficiency of the search procedure for an optimum is the possibility of finding the optimal solution to the problem and convergence (Zhang, 2015).

A common mathematical problem in all fields of science and engineering disciplines, is optimization - finding the best solutions. In nature, optimization algorithms can be deterministic or stochastic. Existing methods for solving optimization problems require huge computational costs, so to solve the problem such as stochastic optimization (Parkinson et al., 2013). A better methods to solve this optimization problem is using meta-heuristics based on iterative improvement, or a population of solutions (both in evolutionary and swarm algorithms). (Kees , 2012).

This paper proposes a novel algorithm for Solving optimization problems that is inspired by the random star explosion and born of new stars that is known as supernova The applicability of supernova has been tested by implementing the algorithm using MATLAB to demonstrate the performance and effectiveness of the SO in solving optimization problems. The algorithm has been tested over 20 will known benchmarks functions from CEC2005 suit, the results have been verified by a comparative study with the state of art optimization algorithms GWO, SCA, MVO, MFO, WOA and also with the first optimization algorithm (PSO). The experimental results and the comparisons with other algorithms show the effectiveness of the proposed optimizer.

The rest of this paper is organized as follows. Section 2, presents the related work of swarm intelligence and bio-inspired algorithms and its achievements in solving optimization algorithms. Section 3 describes the proposed SO algorithm. Section 4 provides experimental results and comparison of SO with other meta-heuristic. Section 5 describe the stability of SO. Section 6 shows the Convergence analysis. Finally, the conclusions and future work are presented in Section 7.

2. Related Work

May algorithms has been inspired from nature, table 1 summarize the most common optimizers inspired from nature. A brief description about some of these algorithms will be described in this section

Table 1. Most common optimizers

Algorithm	Author
Simulated Annealing	Khachaturyan et al., 1979
Genetic Algorithm (GA)	Goldberg, 1989.
Ant Colony Optimization (ACO)	Dorigo, et al., 2006.
Particle Swarm Optimization (PSO)	Kennedy &Eberhart, 1995.
Marriage in Honey Bees Optimization Algorithm (MBO)	Abbass HA, 2001.
Artificial Fish-Swarm Algorithm (AFSA)	Li X, 2003.
Termite Algorithm	Roth M, 2005
Wasp Swarm Algorithm	Pinto PC, 2007.
Monkey Search	Mucherino A & Seref O., 2007.
Bee Collecting Pollen Algorithm (BCPA)	Lu X & Zhou Y., 2008.
Cuckoo Search (CS)	Yang X-S and Deb S., 2009.
Dolphin Partner Optimization (DPO)	Shiqin et al., 2009.
Firefly Algorithm (FA)	Yang et al., 2010.
Bird Mating Optimizer (BMO)	Askarzadeh & Rezazadeh, 2012.
Krill Herd (KH)	Gandomi & Alavi, 2012.
Fruit fly Optimization Algorithm (FOA)	Pan W-T, 2012.
Grey Wolf Optimizer	Mirjalili et al., 2014
A Sine Cosine Algorithm	Mirjalili, 2016

Multi-Verse Optimizer	Mirjalili, et al., 2016
Accelerated PSO	Yang et al.2012
Artificial bee colony	Karaboga and Basturk, 2005
A Bacteria Foraging-Particle Swarm Optimization	Pradhan et al, 2013
Bat algorithm	Xin-She Yang, 2010
Bee colony optimization	Teodorovic et al, 2006
Wolf Search Algorithm	Fong et al. 2016
Bees algorithms	Pham et al.2005
Bees swarm optimization	Drias et al. 2009
Cat swarm	Chu et al.2006
Consultant-guided search	Iordache, 2010
Eagle strategy	Yang and Deb, 2012
Fast bacterial swarming algorithm	Chu et al.2008
Fish swarm algorithm	Li et al., 2002
Polar particle swarm optimizer	Alsayyed and Fakhouri, 2017
artificial fish swarm	Zhai et al., 2012
Good lattice swarm optimization	Su et al. 2007
Glowworm swarm optimization	Krishnanand and Ghose, 2008

The ant algorithm (AI); It all began with the study of behavior real ants. Experiments with Argentine ants, conducted Gossom in 1989 and Deneborg in 1990 served as a starting point for further study of swarm intelligence. of the idea is Marco Dorigo of the University of Brussels, Belgium. It was he who managed for the first time formalizes the behavior of ants and applies a strategy for their behavior to solution of the shortest path problem. Ant Colony Optimization (ACO)(Dorigo , 1992), also known as ant algorithm, is a kind of probability algorithm used to find the optimal path in the graph. It was made by Marco Dorigo in 1992 in his doctoral dissertation "Ant system: optimization by a colony of cooperating agents", inspired by the behavior of the ants in finding the path of the food.

Simulated Annealing (SA) (Khachaturyan et al., 1979) Is a global optimization method that traverses the search space by generating adjacent solutions of the current solution. Advanced adjacency is always acceptable. Low-level adjacency solutions may be accepted based on the probability of differences based on quality and temperature parameters. The temperature parameter is modified with the process of the algorithm to change the nature of the search. Tabu Search (TS) is similar to simulated annealing, and they are traversing the solution by testing the mutations of the independent solution. The simulated annealing algorithm generates only one mutation for an independent solution, and the tabu search produces a lot of variational solutions and moves to the lowest degree of coincidence in the resulting solution. In order to prevent loops and to promote greater progress in the solution space, a taboo list is maintained by partial or complete solution construction. Moving to the element The inclusion of the tabular list is prohibited, and the taboo list is continually updated as the process of dissipating the space is solved. The Hungarian algorithm is a combinatorial optimization algorithm that solves the task allocation problem in polynomial time and promotes the later primitive duality method. American mathematician Harold Kuhn proposed the algorithm in 1955. This algorithm is called the Hungarian algorithm because a large part of the algorithm is based on previous Hungarian mathematicians Dénes Kőnig and Jenő Egerváry's work to create up. (Burkard, 2012) (Martello, 2010)

Differential evolution algorithm (DE) is a post-heuristic algorithm for optimization problems. Essentially, it is a kind of greedy genetic algorithm based on real number coding with gifted ideology (Das, 2011). Differential evolution algorithm is similar to genetic algorithm, including mutation, crossover operation, elimination mechanism, and differential evolution algorithm and genetic algorithm is different from the part of the variation is the difference between the two members of the solution, after the expansion of the current solution by adding members Variable, so the differential evolution algorithm does not need to use the probability distribution to produce the next generation solution

Yang (2009), developed The Cuckoo search algorithm (CSA) a meta-heuristic algorithm in imitating the behavior of cuckoos during the laying of eggs, and namely in the process of forced nesting parasitism. There are species of cuckoos that lay eggs in collective nests along with others cuckoo, although they can throw out competitors' eggs to increase probability of hatching their own chicks. A number of species of cuckoo nests parasitism, laying eggs in the nests of other birds as its kind, and, often, other species. Some birds may conflict with the invasion of the cuckoo that is sometimes such an "invasion" meets rebuff in some birds. For example, if

the host of the nest finds eggs in it another kind, he either throws out these eggs, or simply leaves this nest and build another in a new place.

Janson et al., (2005) proposed hierarchical particle swarm optimizer (HPSO), using dynamic hierarchical tree For the neighborhood structure, the best position of the better particles in the upper layer, the speed of each particle by its own history, the best position and rank tree in the particles, The location of the particles is determined by the best position of the particles. Sequential minimal optimization (SMO) is an algorithm used to solve the problem of optimization problems in support vector machine training. SMO was invented by Microsoft Research Institute's John Plater in 1998, which is currently widely used in the training process of SVM and is implemented in the popular SVM library LIBSVM. In 1998, the SMO algorithm published a stir in the field of SVM research, since previously available SVM training methods had to use sophisticated methods and require expensive third-party quadratic planning tools. The SMO algorithm is a good way to avoid this problem.

a meta-heuristic called Grey Wolf Optimizer (GWO) inspired by grey wolves. The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta, and omega are employed for simulating the leadership hierarchy. Grey wolves mostly search according to the position of the alpha, beta, and delta. They diverge from each other to search for prey and converge to attack prey. In order to mathematically model divergence, we utilize with random values greater than 1 or less than -1 to oblige the search agent to diverge from the prey. This emphasizes exploration and allows the GWO algorithm to search globally. The Whale Optimization Algorithm which mimics the social behavior of humpback whales. The algorithm is inspired by the bubble-net hunting strategy Mirjalili and Lewis, (2016). Sine Cosine Algorithm (SCA) which is recently novel population-based optimization algorithm for solving optimization problems. it creates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution using a mathematical model based on sine and cosine functions. (Mirjalili, 2016). Multi-Verse Optimizer: a nature-inspired algorithm for global optimization algorithm (MVO), the main inspirations of MVO are based on three concepts in cosmology: white hole, black hole, and wormhole. The mathematical models of these three concepts are developed to perform exploration, exploitation, and local search, respectively. (Mirjalili et al., 2015)

Khanesar et al,. (2007), proposed A Novel Binary Particle Swarm Optimization a new interpretation for the velocity of binary PSO was proposed, which is the rate of change in bits of particles. Also a number of benchmark optimization problems are solved using this concept and quite satisfactory results are obtained. Al-Sayyed and Fakhouri, 2017 introduced a novel optimization algorithm called POLARPSO has been that enhances the behavior of PSO and avoids the local minima problem by using a polar function to search for more points in the search space.

Particle Swarm Optimization, (PSO) method of numerical optimization, for the use of which you do not need to know the exact gradient of the optimized function. This method belongs to the group algorithms of swarm intelligence, which describe collective behavior of a decentralized self-organizing system. The systems of swarm intelligence, as a rule, consist of many agents locally interacting with each other and with the environment. The agents themselves are usually rather simple, but all together, locally interacting, create the so-called swarm intelligence. The method was first developed to simulate the social behavior of a flock of birds and fish. As a result of the development of the method, it has been successfully applied to problems finding the extreme points of a function. Niu et al., (2007) uses the master-slave model (master-slaver model), which contains a main group, a number of servants Body, and servant groups, and the main group carries on the search on the basis of the best position provided by the servant. Thi Thanh Binh, (2013) proposed new hybrid particle swarm optimization algorithm for solving MAEDP By proposing new algorithm to solve the MAEDP, called Hybrid Particle Swarm Optimization (HGAPSO).

3. Supernova Optimizer (SO)

In this section we explain the inspiration of the proposed algorithm, the mathematical model and the pseudo code.

3.1 Inspiration

Supernova is a transient astronomical event that occurs during the last stellar evolutionary stages of a massive star's life, whose dramatic and catastrophic destruction is marked by one final titanic explosion. This causes the sudden appearance of a "new" bright star, before slowly fading from sight over several weeks or months. Supernovae may expel much, if not all, of the material away from a star, at velocities up to 30,000 km/s or 10% of the speed of light.

SO is triggered by one of two basic mechanisms: the sudden resignation of nuclear fusion in a degenerate star or the sudden gravitational collapse of a massive star's core. In the first instance, a degenerate white dwarf may accumulate sufficient material from a binary companion, either through accretion or via a merger, to raise its core temperature enough to trigger runaway nuclear fusion, completely disrupting the star.

In the second case, the core of a massive star may undergo sudden gravitational collapse, releasing gravitational potential energy as a supernova. While some observed supernovae are more complex than these two simplified theories, the astrophysical collapse mechanics have been established and accepted by most astronomers for some time.

Due to the wide range of astrophysical consequences of these events, astronomers now deem supernovae research, across the fields of stellar and galactic evolution, as an especially important area for investigation. And this inspired ud to represent supernova theory by mathematical model for the particle movement in the search space of the optimization problem and for generation of new particles as described in section 3.2

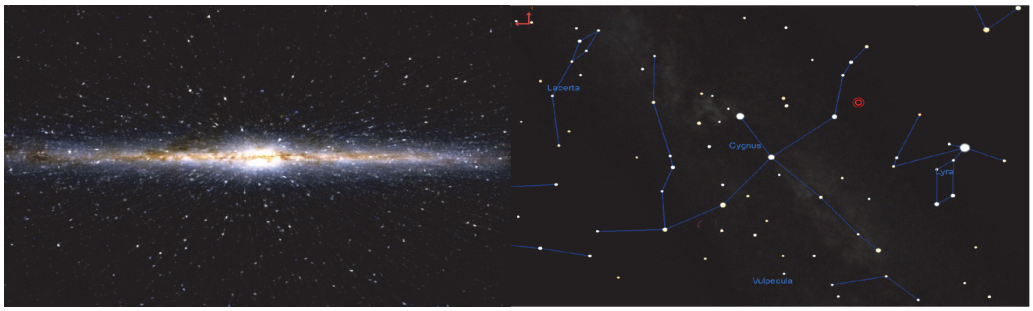


Figure 2. Star Explosion Expected to Create Spectacular Light (space.com, 2017)

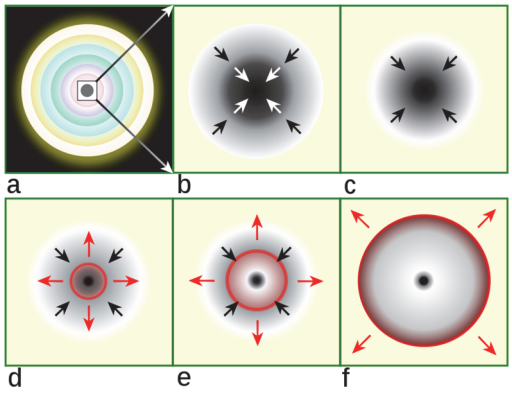


Figure 3. Supernova Core collapse scenario (wikipedia.org, 2017)

As shown in figure 3 within a massive, evolved star (a) the onion-layered shells of elements undergo fusion, forming an iron core (b) that reaches Chandrasekhar-mass and starts to collapse. The inner part of the core is compressed into neutrons (c), causing infalling material to bounce (d) and form an outward-propagating shock front (red). The shock starts to stall (e), but it is re-invigorated by a process that may include neutrino interaction. The surrounding material is blasted away (f), leaving only a degenerate remnant.

3.2 Mathematical Model of Supernova

We chose three main operations of the supernova theory as the inspiration for supernova algorithm first The core of a massive star may undergo sudden gravitational collapse, releasing gravitational potential energy as a supernova and this process has been applied to the swarm of particle as show in equation 1 the particles here represent stars and the operations describe random explosion of stars and random born of new stars, second a degenerate white dwarf may accumulate sufficient material from a binary companion , third raise its core temperature enough to trigger runaway nuclear fusion, completely disrupting the star as shown in equation 2.

GC = GCI (noP, dim, ub ,lb ,SP) …….. Equation 1

Where GC is the Gravitational collapse that will result in new star born, GCI : is the generation of new stars, nop : number of generated stars , dim : the space dimension that the stars will be scattered into , ub : the upper bound of the space dimension , lb : the lower bound of the space dimension , SP: the random position of the generated stars described in equation 2 .

SP = rand()* (ub - lb)+lb + GSP …….... Equation 2

Where SP is the new particle position which represent the local best value at each iteration in supernova algorithm, GSP global supernova value, rand(): is a random function that generate numbers between 0 and 1, ub : the upper bound of the space dimension , lb : the lower bound of the space dimension

SO increases the ability of exploitation and exploration by providing more exploring points. To improve the optimization and search ability we improved the search ability of supernova at each iteration by the inspiration of explosion idea and randomness so that each particle best position may have a near position to it that has better value and this new value will be a start of new particle (star) that when exploited it will give new particles that increase the search space and enhance the exploration feature as described in equation 2 this has extended the search ability to include more points directions of the particle movement which increased the possibility to find the global minima and avoid the local minima problem, rather than getting stuck at one local minima point

After the initialization phase each particle had a velocity vector , position and after, the control parameter has been assigned for each individual, a new particle position will be given to each particle according to best previous solution that were gathered from previous generation. According to equation 3

NP =rand()* Pi (G) ……. Equation 3

Where NP is the best new Position of the, rand( ) is random value between [0,1], Pi (G) is the best value for the particle that has been obtained in the previous generation G,

After calculating the best position we will now update all particle position according the new position and this is performed by re initializing all the particles again according to Equation. 1 and 2

3.2.1 The Proposed Algorithm Consist of Two Phases the Initialization Phase, the Iteration Phase

3.2.1.1 Initialization

Similar to other swarm optimization algorithms for numerical optimization, a the population is represented as a set of real parameter vectors xi = (x1, ..., xD), i = 1, ..., N, where D is the dimensionality of the target problem, and N is the population size. At the beginning of the search, the individual vectors xi in population are initialized randomly. Then, a process vector generation and selection are repeated until termination criterion is encountered.

Each individual will be given a random initial value and position; the best position assemble the best features that is transferred from generation to another. The value rand( ) $\in$ [0, 1] controls the magnitude of the random points that will be added for the best solution.

In case a value for the generated individual outside of [0, 1], it is replaced by the limit value (0 or 1) closest to the generated value. When rand( ) > 1, rand( ) value is truncated to 1, and when rand( ) $\leq$ 0, then it is repeatedly applied to try to generate a valid value.

3.2.1.2 Iteration: Generating New Particles by Using Best Solution (Star Explosion)

In generation G, and at iteration i the elements value in the memory will be updated to a start from a new position near the previously obtained best value NP. At the beginning of the search each particle is initialized randomly. but in Equation.2 the particles will be now initialized and get benefit from the best value (GSP) that has been found so far at each is incremented iteration note that all particles(star) will always regenerated to a new position randomly that will be near the explosion star which is near the best value and this will avoid the local optima problem. And give them new values for exploitation. At each iteration the fitness value will be calculated according the tested bench marked function, the pseudo code of SO algorithm are shown in figure 4

For each star
Initialize star position, energy, supernova global value
END
Return back the particles that go beyond the boundaries of the search
Do
For each star

```
Calculate objective function ( fitness value ) for each star
Update supernova global value
 End
 Choose the particle with the best fitness value
 For each star
 Update star position according equation (3)
 Generate new stars according equation (1)
 Initialize new star position, energy according to equation (2)
Calculate fitness value
 Update global supernova global value
 End
 Update stars position
 End
While maximum iterations or minimum error criteria is not attained
```

Figure 4. Pseudocode of the algorithm

4. Experimental Results and Discussion

The performance of supernova has been evaluated on the CEC2005 Special Session on Real-Parameter Optimization benchmark suite, then we compared supernova to state-of-the-art algorithms including GWO, SCA , MFO, MVO, WOA and PSO. We performed our evaluation on 20 benchmark functions. With dimension D = 30 for F1 to F13, and the maximum number of objective function calls per run was D × 10, 00 (i.e. 30,000). The number of runs per problem was 30, and the average performance and standard deviation of these runs was evaluated. The CEC2005 benchmark set consists of 20 test functions described in table 1. For all of the problems Functions F1 to F5 are Unimodal Functions shown in table 2. F6 to F12 are Multimodal Functions shown in table 2. F13 to F14 are Expanded Functions shown in table 3. Finally, F15 to F20 shown in table 5 are Hybrid Composition Functions which combine multiple test problems into a complex landscape.

In the evaluation we care about Exploitation Feature, Exploration Feature, Avoiding local minima. To test Exploitation Features we use the unimodal functions are suitable for benchmarking exploitation Therefore, these table results illustrated that SO is better than the compared algorithms in terms of optimum exploitation., It can be inferred from the results in table 10 that the SO algorithm have shown a very competitive results in functions F1,F2,F3,F4 and F5 with all other algorithms (GWO, SCA, PSO, MFO, MVO, WOA) , Exploration Feature, since the multimodal functions F8 to F20 have many local optima with the number increasing exponentially with dimension. Then they were suitable to test the exploration behavior of the algorithm. The results of table 11 to 13 of the compared algorithms showed a very strong results of the SO over other algorithms basic the multimodal benchmark functions and fairly competitive results in both the expanded multimodal benchmark functions It can be inferred that SO has a very good exploration feature

Table 2. Bench Mark Functions

Unimodal Functions
F1: Shifted Sphere Function
F2: Shifted Schwefel’s Problem 1.2
F3: Shifted Rotated High Conditioned Elliptic Function
F4: Shifted Schwefel’s Problem 1.2 with Noise in Fitness
F5: Schwefel’s Problem 2.6 with Global Optimum on Bounds
Multimodal Functions: Basic Functions
F6: Shifted Rosenbrock’s Function

F7: Shifted Rotated Griewank's Function without Bounds
F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds
F9: Shifted Rastrigin's Function
F10: Shifted Rotated Rastrigin's Function
F11: Shifted Rotated Weierstrass Function
F12: Schwefel's Problem 2.13
Multimodal Functions: Expanded Functions
F13: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)
F14: Shifted Rotated Expanded Scaffer's F6
Multimodal Functions : Hybrid Composition Functions
F15: Hybrid Composition Function
F16: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum
F17: Rotated Hybrid Composition Function with the Global Optimum on the Bounds
F18: Rotated Hybrid Composition Function
F19: Rotated Hybrid Composition Function with High Condition Number Matrix
F20: Non-Continuous Rotated Hybrid Composition Function

Table 3. Benchmark Functions Mathematical Formulation

Unimodal Functions mathematical formulation

	Formula	**Upper bound**	**Lower bound**	**Dimension**	**F Min**
F1	$f1(x) = \sum_{i=1}^{n} x_i^2$	-100	100	30	0
2	$f2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	-10	10	30	0
3	$f3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_j \right)^2$	-100	100	30	0
4	$f4(x) = \max i\{\lvert x_i \rvert, 1 \le i \le n\}$	-100	100	30	0
5	$f5(x) = \sum_{i=1}^{n=1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	-30	30	30	0
Multimodal Basic Functions					
6	$f6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	-100	100	30	0
7	$f7(x) = \sum_{i=1}^{n} i x_i^4 + random(0,1)$	-1.28	1.28	30	0

8	$f8(x) = \sum_{i=1}^{n} -x_i \sin\sqrt{\lvert x_i \rvert}$ * $\sum_{i=1} i x_i^4 *$ random(0,1) *	-500	500	30	-418.98 29*5
9	$f9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	-5.12	5.12	30	0
10	$f10(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{x=i}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	-32	32	30	0
11	$f11(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	-600	600	30	0
12	$f12(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n=1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < a \end{cases}$	-50	50	30	0
Multimodal expanded Functions					
13	$f13(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	-50	50	30	0
14	$f14(x) = -\sum_{i=1}^{n}\sin(x_i).\left(\sin\left(\frac{ix_i^2}{\pi}\right)\right)^{2m}$, m=10	-65.536	65.536	2	1
Hybrid Composition Functions					

15	$f15(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3 x_4}\right]$	-5	5	4	0.00030
16	$f19(x) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	0	1	3	-3.86
17	$f20(x) = -\sum_{i=1}^{4} C_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	0	1	6	-3.32
18	$f21(x) = -\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T + C_i]^{-1}$	0	10	4	-10.1532
19	$f22(x) = -\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T + C_i]^{-1}$	0	10	4	-10.4028
20	$f20(x) = -\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T + C_i]^{-1}$	0	10	4	-10.5363

4.1 Experimental Parameters

We performed our evaluation on the 20 benchmark functions. The maximum number of iterations performed was 1000. The number of runs per problem was 30, and the average performance and standard deviation of these runs were collected for evaluation purposes. As shown in Table 3 which describes the Experimental parameters of the function.

SUPERNOVA				
Function No.	MEAN	STD	MIN	MAX
F1	0	0	0	0
F2	0	0	0	0
F3	0	0	0	0
F4	0	0	0	0
F5	28.94177	0.041179	28.81195	28.98703
F6	6.407081	0.445589	5.353491	7.274983
F7	7.73E-06	7.79E-06	3.39E-07	3.89E-05
F8	-2937.66	362.795	-3607.38	-2183.93
F9	0	0	0	0
F10	8.88E-16	0	8.88E-16	8.88E-16
F11	0	0	0	0
F12	1.191922	0.250995	0.702527	1.529864
F13	2.869524	0.092223	2.6412	2.989601
F14	7.697852	4.132355	1.992036	12.67051
F15	0.015245	0.021199	0.000384	0.067352
F16	-3.75588	0.10269	-3.85782	-3.43531
F17	-2.42127	0.415247	-3.01385	-1.65572
F18	-2.5158	1.055466	-4.2894	-0.86939
F19	-2.66087	0.942216	-4.30718	-0.93619
F20	-2.60698	0.893182	-4.80274	-1.20992

4.2 Comparison with Other Algorithms

We compared SO with GWO, SCA, PSO and MFO on CEC2005 benchmark set problems, especially the new hybrid functions. The results for D ∈ {30} dimension on the tested functions and the overall results on all 20 functions are shown in Table 4 to Table 6. Due to space, the table only shows the aggregate results of comparing each algorithm vs. SO

Table 4. GWO, SCA MEAN, STD, MIN, MAX values over 30 iterations

GWO					**SCA**			
	MEAN	**STD**	**MIN**	**MAX**	**MEAN**	**STD**	**MIN**	**MAX**
F1	7.81E-94	1.57E-93	1.64E-96	6.49E-93	7.69E-07	2.11E-06	9.52E-11	1.06E-05
F2	1.33E-53	2.83E-53	7.84E-55	1.6E-52	1.01E-07	1.55E-07	3.4E-10	6.22E-07
F3	1.5E-29	8.27E-29	1.89E-36	4.6E-28	322.0492	672.6731	4.012935	3501.587
F4	2.5E-24	1.96E-24	1.1E-25	7.26E-24	3.690787	3.912826	0.215137	18.23089
F5	25.89532	0.805491	24.28707	27.10698	27.83938	0.579548	26.98248	29.58435
F6	0.088737	0.15214	4.15E-06	0.501891	3.598292	0.306655	3.039842	4.282761
F7	0.000164	8.09E-05	3.52E-05	0.000366	0.004204	0.003179	0.000776	0.015886
F8	-6646.72	665.4916	-7899	-5230.34	-4357.93	201.637	-4763.09	-3997.04
F9	0	0	0	0	2.994904	6.971508	2.62E-10	23.05413
F10	9.03E-15	2.09E-15	7.99E-15	1.51E-14	8.066883	9.718148	1.1E-06	20.16582
F11	0.000981	0.003112	0	0.012634	0.028118	0.078641	2.2E-10	0.32762
F12	0.012597	0.013294	1.96E-07	0.06579	0.367418	0.138199	0.254244	0.944247
F13	0.112092	0.098292	3.45E-06	0.412478	1.97509	0.1505	1.628948	2.241467
F14	1.254017	0.676136	0.998004	2.982105	1.064141	0.362246	0.998004	2.982105
F15	0.001014	0.003598	0.000307	0.020363	0.00063	0.00044	0.00031	0.001326
F16	-3.86174	0.002643	-3.86278	-3.8549	-3.85652	0.003136	-3.86255	-3.85447
F17	-3.23261	0.077225	-3.32199	-3.02424	-3.11867	0.074453	-3.27664	-3.00778
F18	-9.82704	1.261739	-10.1531	-5.10059	-5.3486	1.3755	-8.13182	-0.88162
F19	-10.2326	0.947267	-10.4029	-5.1286	-6.14708	1.32284	-8.69155	-4.8016
F20	-10.5362	0.000107	-10.5364	-10.536	-6.28744	1.459733	-9.05847	-4.56591

Table 5. MFO, MVO MEAN, STD, MIN, MAX values over 30 iterations

MFO					**MVO**			
	MEAN	**STD**	**MIN**	**MAX**	**MEAN**	**STD**	**MIN**	**MAX**
F1	333.3333	1825.742	1.65E-06	10000	0.06987	0.01721	0.033928	0.103761
F2	24.9891	19.76896	4.12E-05	70	0.163783	0.031259	0.105083	0.237172
F3	14260.85	9971.811	102.0234	36669.41	3.734672	1.437699	1.190944	6.932981
F4	19.16059	7.465388	7.102247	38.28251	0.299265	0.122061	0.114662	0.61259
F5	18498.67	36398.93	1.988693	90080.18	110.0559	342.5613	24.32757	1909.894
F6	1003.358	3061.683	9.88E-07	10100.25	0.070938	0.020428	0.031562	0.116613
F7	0.475416	1.239681	0.013321	5.408014	0.005609	0.002286	0.002963	0.012472
F8	-9138.87	676.1444	-10490.7	-8035.86	-8140.38	840.3777	-9523.94	-5916.09
F9	111.262	44.8015	27.85886	245.185	97.01212	23.66112	44.79188	130.3767
F10	7.246721	9.158242	0.000549	19.95946	0.289247	0.511412	0.055902	2.131228

F11	6.050675	22.98536	4.1E-06	90.73787	0.249762	0.065966	0.140284	0.393325
F12	0.190133	0.507445	6.29E-07	2.692272	0.673869	0.819071	0.000176	2.700481
F13	0.007054	0.008914	6.36E-07	0.043973	0.012404	0.009557	0.003528	0.053325
F14	0.998004	0	0.998004	0.998004	0.998004	1.52E-12	0.998004	0.998004
F15	0.000895	0.000304	0.000417	0.001489	0.00114	0.003642	0.000308	0.020363
F16	-3.8628	9.36E-16	N/A	N/A	-3.86278	3.39E-08	-3.86278	-3.86278
F17	-3.1876	0.0571	N/A	N/A	-3.26252	0.060492	-3.322	-3.20295
F18	-6.3918	3.3827	N/A	N/A	-7.79302	2.808354	-10.1532	-2.63047
F19	-8.2077	3.5443	N/A	N/A	-9.69695	1.830623	-10.4029	-5.08766
F20	-8.5637	3.2229	N/A	N/A	-9.205	2.487936	-10.5364	-2.80663

Table 6. SUPERNOVA, PSO MEAN, STD, MIN, MAX values over 30 iterations

POLARPSO					**PSO**			
	MEAN	**STD**	**MIN**	**MAX**	**MEAN**	**STD**	**MIN**	**MAX**
F1	7E-122	3.8E-121	5.8E-162	2.1E-120	2.22E-17	3.47E-17	6.90E-21	1.20E-16
F2	8.88E-59	4.79E-58	1.55E-74	2.63E-57	2.80E-08	5.27E-08	4.24E-11	1.90E-07
F3	6.1E-115	3.3E-114	1.2E-145	1.8E-113	1.756151	0.821985	0.408156	3.949604
F4	3.47E-58	1.73E-57	2.71E-73	9.48E-57	0.135487	0.055105	0.057225	0.317909
F5	24.10345	0.459677	23.22311	25.26795	49.30416	30.21335	14.76934	113.3159
F6	0.008699	0.039221	1.06E-09	0.213869	1.78E-17	3.26E-17	5.08E-20	1.50E-16
F7	1.35E-05	1.02E-05	3.26E-07	3.83E-05	0.023137	0.006572	0.012105	0.036585
F8	-3216.19	506.3645	-4148.58	-2127.9	-6874.56	619.3797	-8364.73	-5265.23
F9	0	0	0	0	31.32266	8.231646	11.93951	45.77825
F10	8.88E-16	0	8.88E-16	8.88E-16	1.80E-09	1.61E-09	1.31E-10	6.37E-09
F11	0	0	0	0	0.009686	0.00867	0	0.029518
F12	7.35E-06	2.83E-05	1.44E-10	0.000144	3.29E-19	1.03E-18	1.28E-21	5.42E-18
F13	1.034037	0.942788	0.010989	2.963	2.82E-18	5.69E-18	4.09E-21	2.51E-17
F14	6.895737	4.381316	0.998009	12.67051	1.064272	0.252193	0.998004	1.992031
F15	0.000307	6.18E-19	0.000307	0.000307	0.00053	0.000293	0.000307	0.001054
F16	-3.86278	2.64E-15	-3.86278	-3.86278	-3.86278	2.71E-15	-3.86278	-3.86278
F17	-3.25462	0.059923	-3.322	-3.2031	-3.25066	0.059241	-3.322	-3.2031
F18	-9.98327	0.930764	-10.1532	-5.0552	-8.46602	2.426847	-10.1532	-5.0552
F19	-10.376	0.135834	-10.4029	-9.65956	-9.69698	1.830634	-10.4029	-5.08767
F20	-10.1706	1.300869	-10.5364	-5.12848	-9.82007	1.857554	-10.5364	-5.12848

The obtained results of the comparison for the unimodal benchmark functions F1 to F7 are classified in table 7, multimodal expanded benchmark functions F8 to F12 are classified in table 8, on multimodal Basic benchmark functions F13 to F14 are classified in table 9, Hybrid Composition benchmark functions F15 to F20 are classified in table 10

Table 7. Results of unimodal benchmark functions F1 to F7.

	VS PSO	**VS GWO**	**VS SCA**	**VS MFO**	**VS MVO**
F1	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F2	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)

F3	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F4	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F5	+ (better)	- (worse)	- (worse)	+ (better)	+ (better)
F6	- (worse)	- (worse)	- (worse)	+ (better)	- (worse)
F7	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)

SO showed better results over GWO in 5 functions, SCA in 5 functions, PSO in 6 functions, MVO in 6 functions and MFO in 7 functions .

Table 8. Results of multimodal Basic benchmark functions F8 to F12:

	VS PSO	**VS GWO**	**VS SCA**	**VS MFO**	**VS MVO**
F8	- (worse)	- (worse)	+ (better)	- (worse)	- (worse)
F9	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F10	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F11	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F12	- (worse)	- (worse)	- (worse)	- (worse)	- (worse)

SO showed better results over GWO in 3 functions, SCA in 4 functions, PSO in 3 functions MVO in 3 functions, and MFO in 3 functions.

Table 9. Results of multimodal expanded benchmark functions F13 to F14:

	VS PSO	**VS GWO**	**VS SCA**	**VS MFO**	**VS MVO**
F13	- (worse)	- (worse)	- (worse)	- (worse)	- (worse)
F14	+ (better)	+ (better)	+ (better)	- (worse)	- (worse)

SO showed better results over GWO in 1 functions, SCA in 1 functions, PSO in 1 functions while MFO MVO showed better results over SUPERNOVA in the 2 functions.

Table 10. Results of Hybrid Composition benchmark functions F15 to F20

	VS PSO	**VS GWO**	**VS SCA**	**VS MFO**	**VS MVO**
F15	- (worse)	+ (better)	- (worse)	+ (better)	+ (better)
F16	+ (better)	+ (better)	+ (better)	+ (better)	+ (better)
F17	- (worse)	- (worse)	+ (better)	- (worse)	- (worse)
F18	- (worse)	- (worse)	+ (better)	- (worse)	- (worse)
F19	- (worse)	- (worse)	- (worse)	- (worse)	- (worse)
F20	- (worse)	- (worse)	+ (better)	- (worse)	- (worse)

SO showed better results over GWO in only 2 functions, SCA in 4 functions, PSO in 1 functions and MFO and MVO in 2 functions.

4.3 Unimodal Test functions and Exploitation

The results of Table 7 show that the proposed algorithm is able to provide very competitive results on the unimodal test functions. This testifies that the proposed algorithm has a high exploitation ability.

4.4 Multi-Modal Test Functions and Exploration Analysis

After discussing the exploitation feature of supernova, we are going to discuss the exploration feature of supernova. to discuss this we will refer to the results obtained from multi-model test function table 8, 9. the results of this table shows that the proposed algorithm is able to provide a very good exploration behavior. it may

be observed that this supernova is better than other algorithms on f9, f110, and f11. The results indicate that supernova results are competitive in exploration. exploration always comes with local optima avoidance. in fact, stagnation in local solutions can be resolved by promoting exploration. in addition to the discussion provided in the preceding paragraph, local optima avoidance of supernova is also competitive

4.5 Composite Test Functions

This subsection discusses the balance between exploration and exploitation, due to the difficulty of this set of test functions. The results of supernova algorithm we not very competitive in these set of functions

5. Stability of SO

The standard deviation values for all functions from F1 to F20 shown in Table 11. Except for F8 are all near the zero and this means that SO is stable for the 30 experiments that were performed, the reason for the high standard deviation value of F8 is due to the very large values in the search space; see Figure. 5.

Table 11. standard deviation values for all functions from F1 to F20

	SO STD	PSO STD	GWO STD	SCA STD	MFO STD	MVO STD	POLARPSO STD
F1	0	3.47E-17	1.57E-93	2.11E-06	1825.742	0.01721	3.8E-121
F2	0	5.27E-08	2.83E-53	1.55E-07	19.76896	0.031259	4.79E-58
F3	0	0.821985	8.27E-29	672.6731	9971.811	1.437699	3.3E-114
F4	0	0.055105	1.96E-24	3.912826	7.465388	0.122061	1.73E-57
F5	0.041179	30.21335	0.805491	0.579548	36398.93	342.5613	0.459677
F6	0.445589	3.26E-17	0.15214	0.306655	3061.683	0.020428	0.039221
F7	7.79E-06	0.006572	8.09E-05	0.003179	1.239681	0.002286	1.02E-05
F8	362.795	619.3797	665.4916	201.637	676.1444	840.3777	506.3645
F9	0	8.231646	0	6.971508	44.8015	23.66112	0
F10	0	1.61E-09	2.09E-15	9.718148	9.158242	0.511412	0
F11	0	0.00867	0.003112	0.078641	22.98536	0.065966	0
F12	0.250995	1.03E-18	0.013294	0.138199	0.507445	0.819071	2.83E-05
F13	0.092223	5.69E-18	0.098292	0.1505	0.008914	0.009557	0.942788
F14	4.132355	0.252193	0.676136	0.362246	0	1.52E-12	4.381316
F15	0.021199	0.000293	0.003598	0.00044	0.000304	0.003642	6.18E-19
F16	0.10269	2.71E-15	0.002643	0.003136	9.36E-16	3.39E-08	2.64E-15
F17	0.415247	0.059241	0.077225	0.074453	0.0571	0.060492	0.059923
F18	1.055466	2.426847	1.261739	1.3755	3.3827	2.808354	0.930764
F19	0.942216	1.830634	0.947267	1.32284	3.5443	1.830623	0.135834
F20	0.893182	1.857554	0.000107	1.459733	3.2229	2.487936	1.300869

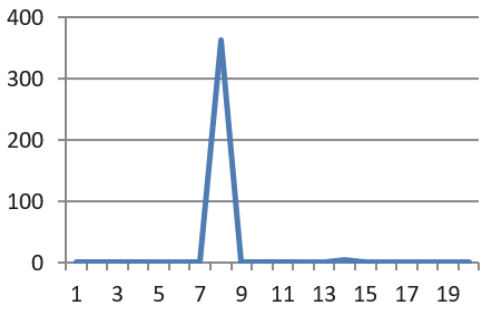
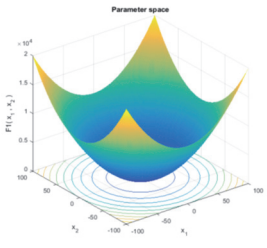
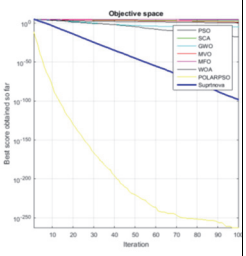
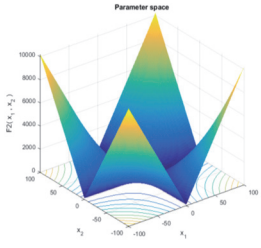
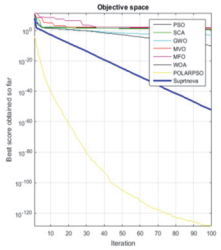
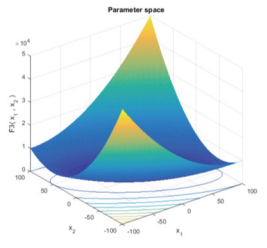
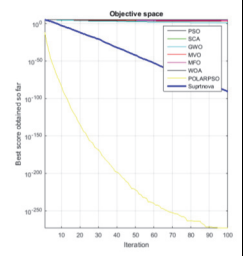
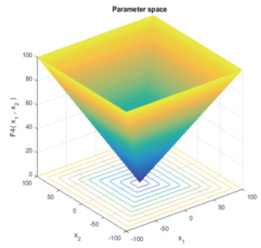
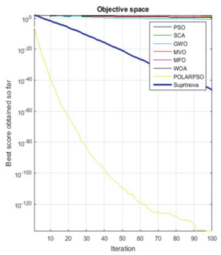
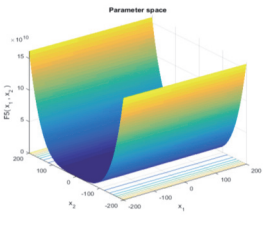
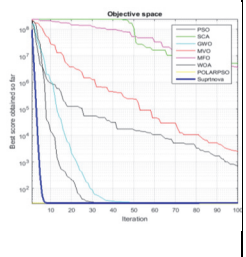
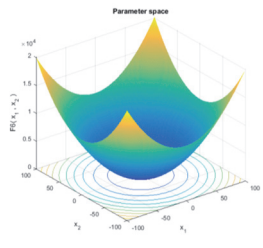
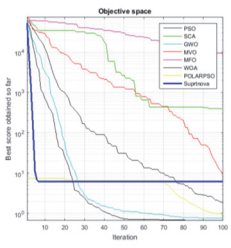
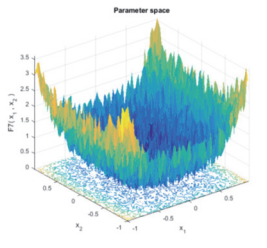
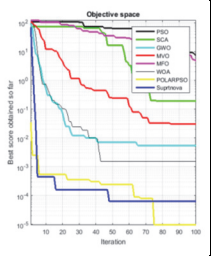
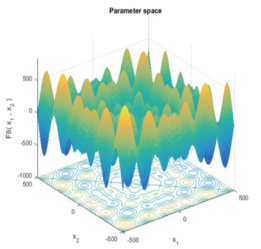
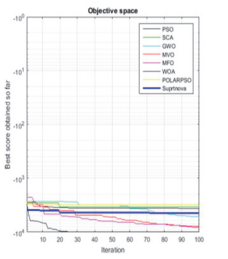
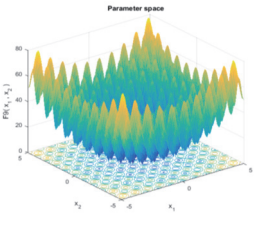
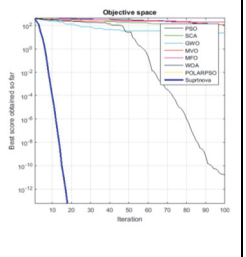
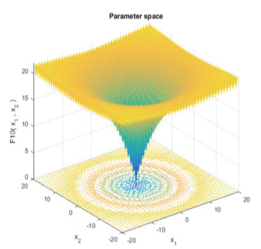
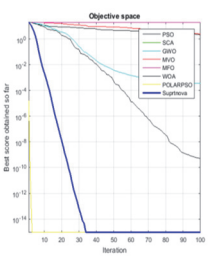
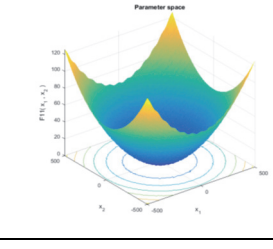
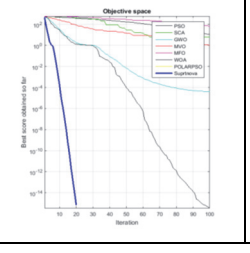
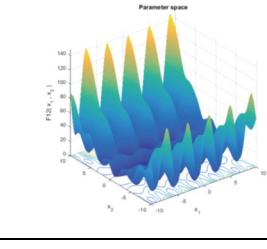
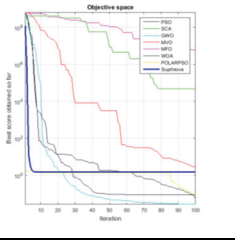


Figure 5. SO standard deviation

6. Convergence Analysis

To confirm the convergence of the proposed algorithm, we provide the convergence curves in Figure 6. This evidences that SO algorithm can successfully improve the fitness of all guarantee finding better solutions as iteration increases. In this is due to that it search for the best global minimum that is generated by the particles that simulate the supernova phenomena

F1			F2	 
F3			F4	 
F5			F6	 
F7			F8	 
F9			F10	 
F11			F12	 

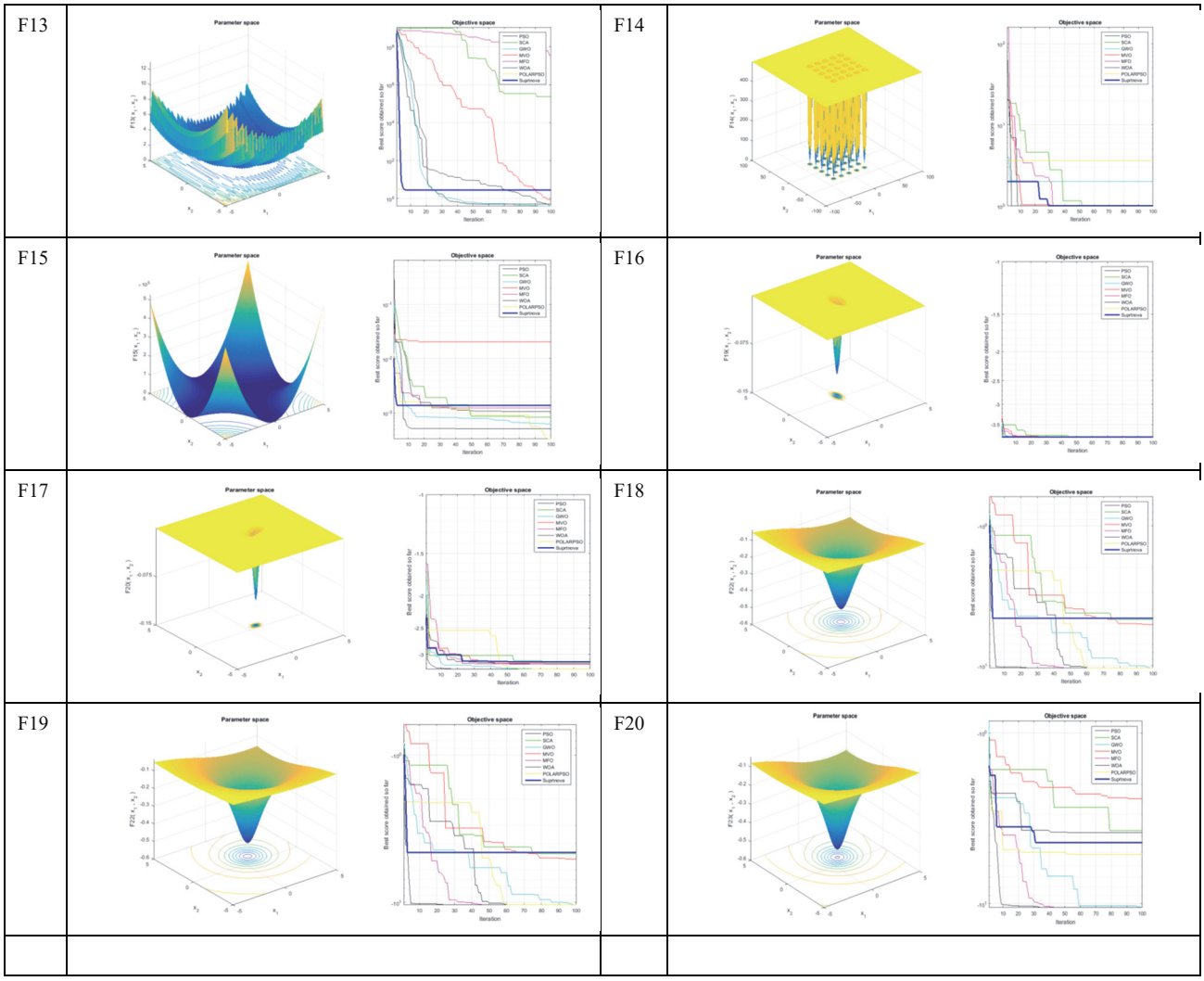


Figure 6. Convergence curve of supernova algorithm in comparison with GWO, SCA, MVO, MFO, PLOARPSO, WOA, and PSO

It can be noticed from figure 6 that supernova optimizer outperformed all of the six optimization algorithms (GWO, SCA, MVO, MFO, WOA, and PSO) and it was the fastest to find the best solution as shown in the Convergence curve chart for each iteration for the bench mark functions from F1 to F5 which represent Unimodal Functions (Shifted Sphere Function, Shifted Schwefel's Problem 1.2 , Shifted Rotated High Conditioned Elliptic Function, Shifted Schwefel's Problem 1.2 with Noise in Fitness, Schwefel's Problem 2.6 with Global Optimum on Bounds) and F7 which is a Multimodal Basic Functions (Shifted Rotated Griewank's Function without Bounds), This testifies that the proposed algorithm has a high exploitation ability.

It can be also observed from figure 2 that supernova optimizer succeed to be the fastest and most effective optimizer for all of the compared algorithms (GWO, SCA, MVO, MFO, WOA, and PSO) except the POLARPSO algorithm when it is tested for the bench mark functions F9- F11 which is from the Multimodal Functions set for Shifted Rastrigin's Function, Shifted Rotated Rastrigin's Function, Shifted Rotated Weierstrass Function respectively

The analysis of results also shows that supernova optimizer showed a competitive and effective performance in finding the best solution that is almost similar to the compared algorithms for the functions F14, F16, F17 which represent more complex problems from the set of Multimodal Expanded Functions and Hybrid Composition Functions that represent the functions of (Shifted Rotated Expanded Scaffer's , Rotated Hybrid Composition Function with the Global Optimum on the Bounds, Rotated Hybrid Composition Function with High Condition Number Matrix , Non-Continuous Rotated Hybrid Composition Function).

7. Conclusion

This paper proposed a novel meta-heuristic optimizer for functions optimization Inspired by the supernova theory. The proposed Supernova Optimizer(SO) was conducted on 20 mathematical benchmark of the cec2005 because they provide a good testing suit for the exploration, exploitation and local optima avoidance of optimization algorithms, we compared our results to the state-of-art GWO, SCA, MFO, MVO, WOA, POLARPSO and PSO meta-heuristic algorithms. The Optimization performance of SO is being evaluated. The experimental results showed that SO incorporated a powerful and competitive way in optimizing the functions when compared to other algorithm as illustrated in table 3 to table 6. It significantly has an improved optimization performance and mostly over the exploitation feature as shown in the Convergence analysis, hence it outperformed most of the compared algorithms.

References

Abbass, H. A. (2001). MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on (Vol. 1, pp. 207-214). IEEE.

Askarzadeh, A., & Rezazadeh, A. (2013). A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer. *International Journal of Energy Research, 37*(10), 1196-1204.

Baskar, S., & Suganthan, P. N. (2004). A novel concurrent particle swarm optimization [A], in Proc. *IEEE Congr. Evol. Comput. [C], 1*, 792–796.

Bergh, F., & Engelbrecht, A. (2004). A cooperative approach to particle swarm optimization [J]. *IEEE Trans. Evol. Comput., 8*(3), 225-239.

Blackwell, T. M., & Bentley, P. J. (2002). Dynamic search with charged swarms. in Proc. Genetic Evol. Comput. Conf. [C], W. B. Langdon et al., Eds., 19–26.

Blackwell, T. M., & Bentley, P. J. (2002). Dynamic search with charged swarms [A]. in Proc. Genetic Evol. Comput. Conf. [C], W. B. Langdon et al., Eds., 19–26.

Brits, R., Engelbrecht, A. P., & van den Bergh, F. (2002). A Niching Particle Swarm Optimizer. in Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning. Singapore: Orchid Country Club, 692-696, 2002.

Burkard, R. E. Dell'Amico, M., & Martello, S. (2012). Assignment Problems (Revised reprint). SIAM, Philadelphia (PA.). ISBN 978-1-61197-222-1

Chu, S. C., Tsai, P. W., & Pan, J. S. (2006). Cat Swarm Optimization. 854-858. https://doi.org/10.1007/11801603_94

Chu, Y., Mi, H., Liao, H. L., Ji, Z., & Wu, Q. H. (2008). A Fast Bacterial Swarming Algorithm For High-dimensional Function Optimization. 2008 IEEE Congress on Evolutionary Computation, CEC 2008. 3135-3140. https://doi.org/10.1109/CEC.2008.4631222.

Das, S., & Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation, 15*, 4-31.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine, 1*(4), 28-39.

Drias, H., Sadeg, S., & Yahi, S. (2005). Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In Cabestany, J., Prieto, A. G., & Sandoval, F. (Eds.), IWANN 2005. LNCS, 3512, 318-325. Springer, Heidelberg (2005)

Dusan, T., Panta, L., & Goran, M. (2006). Bee Colony Optimization: Principles and Applications, Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar, https://doi.org/10.1109/NEUREL.2006.341200

Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (pp. 39-43). IEEE.

Fong, S., Suash, D., Thomas, H., & Li, J. Y. (2016). Eidetic Wolf Search Algorithm with a global memory structure. *European Journal of Operational Research, 254*(1), 19-28.

Goldberg, E. D. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Adison Wesley PC.

https://en.wikipedia.org/wiki/Supernova#/media/File:Core_collapse_scenario.svg

https://www.space.com/35290-star-explosion-expected-earth-sky-2022.html

Khachaturyan, A., Semenovskaya, S., & Vainshtein, B. (1979). Statistical-Thermodynamic Approach to Determination of Structure Amplitude Phases. Sov.Phys. *Crystallography, 24*(5), 519-524.

Krishnanand, K. N., & Ghose, D. S. I. (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Springer, 3*(2), 87-124. https://doi.org/10.1007/s11721-008-0021-5

Langdon, W. B., & Poli, R. (2007). Evolving Problems to Learn About Particle Swarm Optimizers and Other Search Algorithms. *IEEE Transactions on Evolutionary Computation, 11*(5), 561-578. https://doi.org/10.1109/tevc.2006.886448

Li, L. X. et al. (2002). An Optimizing Method based on Autonomous Animals: Fish Swarm Algorithm. presented at the Proc. of Systems Engineering Theory & Practice.

Li, X. (2004). Adaptively choosing neighborhood bests using species in a particle swarm optimizer for multimodal function optimization [A]. in Proc.Genetic Evol. Comput. Conf., 105-116.

Li, X. L. (2003). A new intelligent optimization-artificial fish swarm algorithm. Doctor thesis, Zhejiang University of Zhejiang, China.

Liang, J. J., & Suganthan, P. N. (2005). Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search. in Proceedings of the 2005 IEEE Congress on Evolutionary Computation, 2. IEEE Press, 522–528.

Løvbjerg, M., & Krink, T. (2002). Extending particle swarms with self-organized criticality [A]. in Proc. *IEEE Congr. Evol. Comput. [C], 2*, 1588-1593.

Lu, X., & Zhou, Y. (2008, September). A novel global convergence algorithm: bee collecting pollen algorithm. In International Conference on Intelligent Computing (pp. 518-525). Springer Berlin Heidelberg.

Lugmayr, A., Stockleben, B., Zou, Y., Anzenhofer, S., & Jalonen, M. (2013). Applying "Design Thinking" in the context of media management education. *Multimedia Tools and Applications, 71*(1), 119-157. ISSN 1380-7501.

Martello, S. (2010). Jeno Egerváry: from the origins of the Hungarian algorithm to satellite communication. *Central European Journal of Operations Research, 18*, 47-58.

Maurice, C. (2017). Particle Swarm Optimization [M]. International Scientific and Technical Encyclopedia.

Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120-133.

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.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61.

Mucherino, A., & Seref, O. (2007, November). Monkey search: a novel metaheuristic search for global optimization. In O. Seref, O. E. Kundakcioglu, & P. Pardalos (Eds.), *AIP conference proceedings* (Vol. 953, No. 1, pp. 162-173). AIP.

Mucherino, A., & Seref, O. (2007, November). Monkey search: A novel metaheuristic search for global optimization. In O. Seref, O. E. Kundakcioglu, & P. Pardalos (Eds.), AIP conference proceedings (Vol. 953, No. 1, pp. 162-173). AIP.

Niu, B., Zhu, Y. L., & He, X. X. et al. (2007). A multi-swarm optimizer based fuzzy modeling approach for dynamic systems processing. Neurocomputing, 1-13.

Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems, 26*, 69-74.

Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2005). The bees algorithm. Technical note. Manufacturing Engineering Centre, Cardiff University, UK, 1-57.

Pinto, P. C., Runkler, T. A., & Sousa, J. M. (2007, April). Wasp swarm algorithm for dynamic MAX-SAT problems. In International Conference on Adaptive and Natural Computing Algorithms (pp. 350-357).

Pinto, P. C., Runkler, T. A., & Sousa, J. M. (2007, April). Wasp swarm algorithm for dynamic MAX-SAT problems. In International Conference on Adaptive and Natural Computing Algorithms (pp. 350-357).

Pradhan, R., Kabat, M. R., & Sahoo, S. P. (2013). A Bacteria Foraging-Particle Swarm Optimization Algorithm for QoS Multicast Routing. In: Panigrahi B.K., Suganthan P.N., Das S., Dash S.S. (eds) Swarm, Evolutionary, and Memetic Computing. SEMCCO 2013. Lecture Notes in Computer Science, 8297. Springer, Cham.

Rizik M. H. Al-Sayyed, Hussam N. Fakhouri, & Ali, Rodan (2017). Colin Pattinson, Polar Particle Swarm Algorithm for Solving Cloud Data Migration Optimization Problem. *Modern Applied Science, 11*(8), 2017. ISSN 1913-1844 E-ISSN 1913-1852. Published by Canadian Center of Science and Education.

Rizik, A., Hussaam, F., Ali, R., & Colin, P. (2017). Particle Swarm Algorithm for Solving Cloud Data Migration Optimization Problem. *Modern Applied Science, Polar, 11*(8), 98.

Roth, M. (2005). Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks.

Serban Iordache SCOOP Software GmbH, Köln, Germany. (2010). Consultant-guided search: a new metaheuristic for combinatorial optimization problems, Proceeding GECCO '10 Proceedings of the 12th annual conference on Genetic and evolutionary computation Pages 225-232, Portland, Oregon, USA — July 07 - 11, 2010 ACM New York, NY, USA ©2010 ISBN: 978-1-4503-0072-8. https://doi.org/10.1145/1830483.1830526

Shiqin, Y., Jianjun, J., & Guangxing, Y. (2009, May). A dolphin partner optimization. In Intelligent Systems, 2009. GCIS'09. WRI Global Congress on (Vol. 1, pp. 124-128). IEEE.

Stefan, J., & Martin, M. (2005). A hierarchical particle swarm optimizer and its adaptive variant [J]. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 35*(6), 1272-1282.

Su, S. B., Wang, J., Fan, W. K., & Yin, X. B. (2007). Good Lattice Swarm Algorithm for Constrained Engineering Design Optimization. 2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2007. https://doi.org/10.1109/WICOM.2007.1575

Yang, X. S., & Deb, S. (2010). Eagle strategy using Levy walk and firefly algorithms for stochastic optimization. in: J.R. Gonzalez et al. (Eds.), Nature Inspired Cooperative Strategies for Optimization. *NICSO, 284*, 101-111.

Yang, X. S., & He, L. X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation Archive, 5*(3), 141-149.

Yang, X. S., Deb, S., & Fong, S. (2011). Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications, in: NDT2011, CCIS 136, Springer, pp. 53-66 (2011).

Zhai, Y. K., & Xu, Y. (2012). A novel artificial fish swarm algorithm based on multi-objective optimization, ICIC'12 Proceedings of the 8th international conference on Intelligent Computing Theories and Applications, Pages 67-73 Huangshan, China Springer-Verlag Berlin, Heidelberg ©2012 ISBN: 978-3-642-31575-6. https://doi.org/10.1007/978-3-642-31576-3_9

Zhang, W., Liu, Y., & Clerc, M. (2003). An adaptive PSO algorithm for reactive power optimization. in Proc. 6th Int. Conf. Advances in Power System Control. *Operation and Management,* 302-307.

Zhang, Y. (2015). A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering, 931256.