Computers, Materials & Continua
DOI:10.32604/cmc.2021.013648
Article

Rock Hyraxes Swarm Optimization: A New Nature-Inspired Metaheuristic Optimization Algorithm

Belal Al-Khateeb[1,*], Kawther Ahmed[2], Maha Mahmood[1] and Dac-Nhuong Le[3,4]

[1]College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq
[2]General Directorate of Scientific Welfare, Ministry of Youth and Sport, Baghdad, Iraq
[3]Institute of Research and Development, Duy Tan University, Danang, 550000, Vietnam
[4]Faculty of Information Technology, Duy Tan University, Danang, 550000, Vietnam
[*]Corresponding Author: Belal Al-Khateeb. Email: belal@computer-college.org
Received: 30 August 2020; Accepted: 01 October 2020

Abstract: This paper presents a novel metaheuristic algorithm called Rock Hyraxes Swarm Optimization (RHSO) inspired by the behavior of rock hyraxes swarms in nature. The RHSO algorithm mimics the collective behavior of Rock Hyraxes to find their eating and their special way of looking at this food. Rock hyraxes live in colonies or groups where a dominant male watch over the colony carefully to ensure their safety leads the group. Forty-eight (22 unimodal and 26 multimodal) test functions commonly used in the optimization area are used as a testing benchmark for the RHSO algorithm. A comparative efficiency analysis also checks RHSO with Particle Swarm Optimization (PSO), Artificial-Bee-Colony (ABC), Gravitational Search Algorithm (GSA), and Grey Wolf Optimization (GWO). The obtained results showed the superiority of the RHSO algorithm over the selected algorithms; also, the obtained results demonstrated the ability of the RHSO in convergence towards the global optimal through optimization as it performs well in both exploitation and exploration tests. Further, RHSO is very effective in solving real issues with constraints and new search space. It is worth mentioning that the RHSO algorithm has a few variables, and it can achieve better performance than the selected algorithms in many test functions.

Keywords: Optimization; metaheuristic; constrained optimization; rock hyraxes swarm optimization; RHSO

1 Introduction

Nature is full of social behaviors that work to accomplish many different tasks. Naturally, all these behaviors coexist in different environments, and their main goal is to survive, but there is what makes them work in the form of swarms, groups, flocks, and colonies because of hunting and mobility and defend themselves. The food search is also essential for social interactions that allow them to complete their lives and reproduction. Another reason for the swarm for some creature is navigation. Birds are the best examples of such behaviors, which migrate intercontinental in flocks appropriately [1,2].

Metaheuristic optimization techniques are used extensively to solve most of the problems; this made it very common to be used as primary methods of acquiring the optimum results of optimization problems. Especially, few of them like Genetic Algorithm (GA) [3], Grey Wolf Optimization (GWO) [4], Ant Colony Optimization (ACO) [5], Slap Swarm Optimization (SSO) [6], Artificial-Bee-Colony (ABC) [7] and Particle Swarm Optimization (PSO) [8] are very known and used in broad distinct fields. Metaheuristics have become very common for several reasons: metaheuristics are very clear. They are fundamentally inspired by physical phenomena, animal behaviors, or evolutionary concepts. Metaheuristics are easy to use, which allow researchers to mimic natural behavior, enhance or produce new metaheuristics and hybridize more than one metaheuristics. Furthermore, this simplicity helps scientists to acquire information rapidly and adapt them for problems solving. Also, flexibility indicates the applicability of metaheuristics to various problems without notable changes in the algorithm's structure. Metaheuristics can be easily applied to various problems. Thus, all a designer need is knowing and understanding how to represent his problems. Also, most metaheuristics have derivation-free mechanisms. In inequality to gradient-based optimization approaches, metaheuristics enhance problems randomly. Metaheuristics deals with problems stochastically. The optimization process starts with random solutions, and there is no need to calculate the derivative of search spaces to find the optimal solution. Finally, metaheuristics are more capable than classical optimization strategies in managing local optima. This is due to the random nature of metaheuristics, which permit them to overcome local solutions inactivity and widely search the whole search area.

Swarm Intelligence (SI) was proposed in 1993 [9] and has been defined by Bonabeau et al. [3] as "the emergent collective intelligence of groups of simple agents". The SI inspiration techniques are created primarily from natural habitations, herds, and flock. There are many examples of SI popular techniques, such as PSO [8], ABC [7], and ACO [5].

Regardless of the variation among metaheuristics, usually, they have a common trait, which is dividing the search process into the exploration and exploitation phases [10–12]. The exploration phase indicates the investigating process of search space as widely as possible. An algorithm must have the stochastic operators globally and randomly search within the search space to support this phase. In contrast, exploitation indicates local search capability about the concerned regions taken from the previous phase (exploration). The appropriate balance of these phases is seen as a challenging task because of the metaheuristics stochastically nature. This paper introduces a new SI technique that is inspired by the behavior of Rock Hyraxes swarms. This algorithm has few parameters that make it easy to implement; the proposed algorithm can also balance between exploration and exploitation phases, making it appropriate for solving many optimization problems. The rest of the paper is organized as follows: Section 2 shows the relevant works. Section 3 presents the Rock Hyraxes Swarm Optimization (RHSO) Algorithm inspiration and mathematical model. Section 4 presents the results and discussion of test functions. Finally, Section 5 concludes the work and suggests several future research directions.

2 Literature Review

There are three major categories of metaheuristics; those are Evolutionary Algorithm (EA), Physics-Based (PB), and Swarm Intelligence (SI) algorithms. EAs, which inspired by the genetic and evolutionary behaviors of creatures. In this branch, the most popular algorithm is the Genetic Algorithm (GA). GA was proposed by Holland in 1992 [13] to simulate Darwinian evolution concepts. Goldberg [14] extensively investigated the engineering applications of GA. Typically, the development is finished by evolving associate initial random solution in EAs. Each new population is created by the cross over and mutation of the previous generation solutions. Since the best solution can higher generate the new population, which most likely creates this new population higher than the population within the previous generation(s), this may make the initial random population converges to the best solutions over generations. An example of EAs are Differential Evolution (DE) [15], Evolutionary Programming (EP) [16], Evolution Strategy (ES) [17,18], Genetic Programming (GP) [19] and Biogeography Based Optimizer (BBO) [20]. Simon initially produced the BBO algorithm rule in 2008 [20]. The elemental set up of this algorithm program is galvanized by biology science that refers to biological organisms' study in geographical distribution.

In physics-based techniques, the development algorithms generally simulate the physical rules. Among the physics-based metaheuristic algorithms are Gravitated Native Search (GLSA) [21], Big Bang Big Crunch (BBBC) [22], Gravitated Search Algorithm (GSA) [23], Charged System Search (CSS) [24], Artificial Reaction Improvement Rule [25], Part rule [26], Ray improvement [27] rule, Small World improvement rule [28], Galaxy based Search Algorithm (GbSA) [29] and arced area improvement [30]. Those algorithms' method is completely unlike EAs as it uses a random set of search spaces that can communicate and move throughout search space per physical rules. This movement is enforced, for instance, exploitation gravitation, inertia force, magnetic force, weights, ray casting, and many others.

GSA is an example of a physics-based algorithm rule at that its basic physical theory is impressed by Newton's law of universal gravitation. The GSA performs a search by employing a group of agents with lots of proportional to the value of fitness performance. Throughout iteration, the gravitated forces between them interest the masses in each another [23].

Mirjalili et al. [4] propose a new metaheuristic called Grey Wolf Optimizer (GWO) inspired by grey wolves' behavior. The GWO algorithmic simulates the leadership hierarchy and looking mechanism of grey wolves in nature. Four grey wolves, such as alpha, delta, beta, and omega, are used to simulate the leadership hierarchy. Additionally, GWO implements three significant steps of looking; those are hunting, attacking prey, and peripheral prey.

SI methods are algorithms that simulate the social behavior of flocks, herds, swarms, or other creatures. The mechanism is near to the performance of the physics-based mostly rule; but the search agents explore looking space and simulate creatures' collective and social intelligence. PSO proposed by Kennedy and Eberhart [8] is one of the famous SI techniques samples. PSO is inspired by the birds flocking social behavior. PSO uses multiple particles that follow the most effective particle's position and their own to data best-obtained positions.

The last example of SI is the Bees' swarming. Artificial Bee Colony Algorithm introduced by Dervis Karaboga et al. [31] in (2007), ABC is an associate optimization algorithm supported the intelligent behavior of bee swarm. ABC testing on tests and compare it with different algorithms like PSO and GA. The results showed that the ABC algorithm outperforms the opposite algorithms.

From all the above, it can be seen that no one metaheuristics algorithm can solve all kinds of problem domains; therefore, there is always a need for new algorithms that can address many types of problem domains.

3 Rock Hyraxes Swarm Optimization (RHSO)

The mathematical model of the proposed method, together with its inspiration, is discussed in this section.

3.1 Inspiration

Rock hyrax (*Procavia capensis*) is a small furry mammal that lives in rocky landscapes across sub-Saharan Africa and along the Arabian Peninsula coast. These mammals live in colonies usually dominated by a single male who aggressively defends his territory; the colonies sometimes up to 50 individuals. They sleep together, look for food together, and even raise their babies together (who then all play together). There are three types of hyrax, two are known as the rock (or bush) hyrax and the third as tree hyrax. In the field, it is sometimes difficult to differentiate between them [32,33].

Rock hyraxes live in areas that vary widely in ambient temperatures and provide adequate water and feed. Low metabolic rates and transparent body temperatures may have contributed to the successful extraction of the rocky protrusions isolated through their distribution [34].

The rock hyrax feeds every day in a circle formation, with their head pointing to the outside of the circle to keep an eye out for predators, such as leopards, hyenas, jackals, servals, pythons, and the Verreaux's eagle and black eagle. When the group is feeding or basking, either the breeding male or a female (Leader) will keep a lookout from a high rock or branch and will give a sharp alarm or call if danger threatens, at which point the group will scurry for cover [35].

3.2 Mathematical Model and Algorithm

Rock hyraxes start taking solar baths for several hours and sharing places to live together; they looking for food together in a distinctive way: forming a circle with different dimensions and angles to get their food. When they find food, the Leader takes a higher place to find food and protect each other from predatory animals.

To mathematically model the Rock hyrax swarm, the population first consists of Leader and members. As mentioned above, the Leader chooses the higher and best place to observe the rest of the group.

The Leader then updates his location based on his old location using the following equation:

$$\boldsymbol{Leader} = \boldsymbol{r_1} * \boldsymbol{x}\left(\boldsymbol{Leader_{pos}}, \boldsymbol{j}\right) \tag{1}$$

where $\boldsymbol{r_1}$ refers to a random number between [0,1], $\boldsymbol{x}$ refers to the previous position of Leader, $\boldsymbol{Leader_{pos}}$ represent "old position of the leader" and $\boldsymbol{j}$ refers to "refers to each **diminution**." After updating the leader position, all members (or search agents) update their position based on their position.

$$\boldsymbol{x}(\boldsymbol{i},\boldsymbol{j}) = (\boldsymbol{x}(\boldsymbol{i},\boldsymbol{j}) - (\boldsymbol{circ} * \boldsymbol{x}(\boldsymbol{i},\boldsymbol{j}) + \boldsymbol{Leader})) \tag{2}$$

where $\boldsymbol{circ}$ refers to circular motion and try to mimic the circle system, it is calculated as:

$$n_1 = r_2 * \cos(ang)$$

$$n_2 = r_2 * \sin(ang)$$

$$circ = sqrt\left(n_1 \hat{2} + n_2 \hat{2}\right)$$

where $\boldsymbol{r_2}$ refers to the radius, and it is a random number between [0, 1], $\boldsymbol{ang}$ is a random number between [0, 360], and it refers to the angle of a move. The $\boldsymbol{ang}$ also updated in every generation, and this update depending on the lower and upper bounds of the variables, where ***lb*** and ***ub*** are the lower and upper bands of the random number generation.

$$\boldsymbol{dalta} = \boldsymbol{random}[\boldsymbol{lb}, \boldsymbol{ub}] \tag{3}$$

$$\boldsymbol{ang} = \boldsymbol{ang} + \boldsymbol{dalta} \tag{4}$$

The angle (***ang***) can be kept within the specified range by making it either equal to 360 if the value of the output becomes greater than 360, or equal to 0 if it becomes less than 0.

The pseudo-code of the RHSO algorithm is shown in Algorithm 1. The RHSO starts optimization by creating random solutions in the explorative mode and calculating their fitness. Depending on their fitness, it identifies the best fitness as Leader; this represents switching from explorative mode to local exploitation mode that focuses on the promising regions when global optimal may be in a close place. Later, the Leader represents the best solution for optimization problems. The search agents start again with another set of explorative moves and subsequent turn into a new exploitation stage. The Leader updates his position based on Eq. (1), while the remaining members update their positions according to the Leader's position, as shown in Eq. (2). Calculating the fitness of each search agent and selecting the best one as a leader. This process will continue throughout iterations; when reaching the stopping condition, the process returned the Leader as the best approximation for the optimization problem's best solution.

Algorithm 1: Rock Hyrax Swarm Optimization

```
Initialize the population of Rock Hyrax N member.
Calculate the fitness of each search agent.
Leader = the best search agent.
t = 1.
While (t < Max number of iterations)
        Update Leader position, according to Eq. (1).
        Update the position of each search agent, according to Eq. (2).
        Calculate the fitness of each search agent.
        Select the best member of the population as a Leader.
        Update the angle, according to Eqs. (3) and (4).
        t = t + 1.
End while
        Return Leader as the best solution.
```

4 Results and Discussion

RHSO algorithm is evaluated by using 48 benchmark functions. Some of those functions are standard functions that are used in researches. These functions are chosen to be able to show the performance of RHSO and also to compare it with some known algorithms. The selected 48 test functions are shown in Tabs. 1 and 2, where D means the function's dimension, range is the function's search space limits, and Opt is the optimal value. The selected functions are unimodal or multimodal benchmark minimization functions. Unimodal test functions have a single optimum value; thus, they can benchmark an algorithm's convergence and exploitation. While multimodal test functions have more than one optimum value, making them more challenging than unimodal. An algorithm should avoid all the local optima to approach and approximate the global optimum.

So, exploration and local optima avoidance of algorithms can be benchmarked by multimodal test functions.

Table 1: Unimodal benchmark functions

Equation	Test name	D	Range	Opt
$f_1(x) = \sum_{i=1}^{n} x_i^2$	Sphere	30	100, −100	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	Schwefel 2.22	2	100, −100	0
$f_3(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	Schwefel 2.21	2	100, −100	0
$f_4(x,y) = -200e^{-0.2\sqrt{x^2+y^2}}$	Ackley 2	2	32, −32	−200
$f_5(x) = {x_1}^2 + {x_2}^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	Bohachevskyn N.1	2	100, −100	0
$f_6(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x^2 - 5)^2$	Booth	2	10, −10	0
$f_7(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$	Zakharov	2	5.12, −5.12	0
$f_8(x) = 0.26\left({x_1}^2 + {x_2}^2\right) - 0.48x_1x_2$	Matyas	2	10, −10	0
$f_9(x) = \sum_{i=1}^{n} x_i^{10}$	Schwefel 2.23	2	100, −100	0
$f_{10}(x) = \sum_{i=1}^{n} \lvert x_i\rvert$	Schwefel 2.20	2	100, −100	0
$f_{11}(x) = 0.5 + \frac{sin^2\left(x_1^2 - x_2^2\right)^2 - 0.5}{\left[1 + 0.001\left(x_1^2 - x_2^2\right)\right]^2}$	Schaffer N1	2	100, −100	0
$f_{12}(x) = \sum_{i=1}^{d}[(\sum_{j=1}^{d} x_j^i) - b_i]^2$	Power sum	4	4, 0	0
$f_{13}(x) = 0.5 + \frac{sin^2\left(x_1^2 - x_2^2\right) - 0.5}{\left[1 + 0.001\left(x_1^2 - x_2^2\right)\right]^2}$	Schaffer 2	2	100, −100	0
$f_{14}(x) = \left[e^{-\sum_{i=1}^{n}\left(\frac{x_i}{\beta}\right)^{2m}} - 2e^{-\sum_{i=1}^{n}{x_i}^2}\right] \cdot \prod_{i=1}^{n} cos^2 x_i, m = 5$	Xin-She Yang 3	30	20, −20	−1
$f_{15}(x) = \sum_{i=1}^{d} ix_i^2$	Sum Squares	2	10, −10	0
$f_{16}(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2}}{0.5\left(x_1^2 + x_2^2\right) + 2}$	Droup Wave	2	5.12, −5.12	−1
$f_{17}(\mathrm{x}) = \sum_{i=1}^{\frac{d}{4}}\left(x_{4i-3} + 10x_{4i-2}\right)^2 + 5\left(x_{4i-1} - x_{4i}\right)^2 + \left(x_{4i-2} - x_{4i-1}\right)^4 + 10\left(x_{4i-3} + 10x_{4i}\right)^4$	Powell	2	4, −5	0
$f_{18}(x) = \sum_{i=1}^{d} \lvert x_i\rvert^{i+1}$	powell_sum	2	1, −1	0
$f_{19} = \sum_{i=1}^{d}\sum_{j=1}^{i} x_j^2$	Rotated hyper-ellipsoid	2	65.536, −65.536	0
$f_{20} = x_1^2 - x_1x_2 + x_2^2$	Rotated ellipse 2	2	500, −500	0
$f_{21} = \sum_{i=1}^{d} \epsilon_i \lvert x_i\rvert^i$	Xin-She Yang 1	2	5, −5	0
$f_{22} = 0.5 + \frac{sin^2\left(x_1^2 x_2^2\right) - 0.5}{\left[1 + 0.001\left(x_1^2 x_2^2\right)\right]^2}$	schaffer2	2	100, −100	0

Table 2: Multimodal benchmark functions

Equation	Test name	Type	D	Range	Opt
$f_{23}(x) = -20\exp\left(-0.2\sqrt{\sum_{i=1}^{n} x_i{}^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	Ackley	N	2	10, −10	0
$f_{24}(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	Quartic	N	10	1.28, −1.28	0
$f_{25}(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	Six-Hump camel	N	2	5, −5	−1.0316
$f_{26}(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)cos(x_1) + s$	Branin		2	15, −5	0.3979
$f_{27} = 10d + \sum_{i=1}^{d} x_i^2 - 10\cos(2\pi x_i)$	Rastrigin	N	2	5.12, −5.12	0
$f_{28}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	Hartmann 3-D	F	3	1, 0	−3.8628
$f_{29}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	Three-Hump camel	N	2	5, −5	0
$f_{30}(x) = -200e^{-0.2\sqrt{x^2+y^2}} + 5e^{\cos(3x)+\sin(3y)}$	Ackley 3		2	32, −32	−195.629
$f_{31}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	Bohachevskyn N.2	N	2	10, −10	0
$f_{32}(x) = \sin(x)\, e^{(1-\cos(y))^2} + \cos(x)\, e^{(1-\sin(x))^2} + (x-y)^2$	Brid	N	2	2 pi, −2 pi	−106.7645
$f_{33}(x) = \left(\left\|\sin(x_1)\sin(x_2)\, exp\left(\left\|100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\|\right)\right\| + 1\right)^{0.1}$	Cross in tiny	N	2	10, −10	−2.06261
$f_{34}(x) = -\frac{sin^2(x-y)\, sin^2(x+y)}{\sqrt{(x^2+y^2)}}$	Keane	N	2	10, 0	−0.6737
$f_{35}(x) = -\sum_{i=1}^{d}(x_i - 1)^2 - \sum_{i=2}^{d} x_i x_{i-1}$	Trid	F	6	36, 36	−50
$f_{36}(x) = \frac{sin(10\pi x)}{2x} + (x-1)^4$	Gramacy & Lee	F	1	2.5, 0.5	−0.869
$f_{37} = \left\|x_1^2 + x_2^2 + x_1x_2\right\| + \|\sin(x_1)\| + \|\cos(x_2)\|$	Bartels Conn	N	2	500, −500	1
$f_{38}(x) = -\sum_{i=1}^{d} sin(x_i)\, sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	Michalewics	N	2	2.21, 1.57	−1.8013
$f_{39}(x) = \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Griewank	N	30	600, −600	0
$f_{40}(x) = \left(\sum_{i=1}^{n} sin^2(x_i) - e^{-\sum_{i=1}^{n} x_i{}^2}\right) e^{-\sum_{i=1}^{n} sin^2\sqrt{\|x_i\|}}$	Xin-She Yang 4	N	30	10, −10	−1
$f_{41}(x) = \sum_{i=1}^{n}\|x_i\sin(x_i) + 0.1x_i\|$	Alpine 1	N	30	10, −10	0
$f_{42}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	Bohachevsky 3	F	2	100, −100	0
$f_{43}(x) = x_1^2 + 2x_2^2 + 25\left[sin^2(x_1) + sin^2(x_2)\right]$	EggCrate	F	2	5, −5	0
$f_{44}(x) = 1 + sin^2(x_1) + sin^2(x_2) - 0.1e^{\left(-x_1^2-x_2^2\right)}$	Price 2	F	2	10, −10	0.9
$f_{45}(x) = \left(2x_1^3x_2 - x_2^3\right)^2 + \left(6x_1 - x_2^2 + x_2\right)^2$	Price 4	F	2	50, −50	0
$f_{46}(x) = -3803.84 - 138.08x_1 - 232.92x_2 + 128.08x_1^2 + 203.64x_2^2 + 182.25x_1x_2$	Quadratic	F	2	10, −10	−3.8737e + 03
$f_{47} = \left(x_1^2 + x_2^2 - 2x_1\right)^2 + 0.25x_1$	Zett1	F	2	5, −1	−3.80E−03
$f_{48} = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{d} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{d} x_i^2}$	Salomon	N	2	100, −100	0

Table 3: Results of unimodal benchmark functions

Function	RHSO		ABC		PSO	
	Average	Std	Average	Std	Average	Std
F1	0	0	1.252987E−03	8.111558E−04	2.540288E−08	7.983156E−08
F2	0	0	2.537476E−04	1.524380E−04	1.053400E−55	5.146430E−55
F3	0	0	1.662766E−03	1.098175E−03	2.028025E−51	1.100532E−50
F4	−200	0	−199.9998	1.017095E−04	−200	0
F5	0	0	1.422351E−05	1.412759E−05	1.665335E−16	5.887396E−16
F6	0	0	4.518208E−07	5.788988E−07	6.968271E−30	2.475261E−29
F7	0	0	1.841699E−09	1.699341E−09	4.08255E−104	2.08070E−103
F8	0	0	1.329306E−07	1.364219E−07	6.774106E−66	3.710326E−65
F9	0	0	2.946184E−31	9.239415E−31	0	0
F10	0	0	5.618590E−04	2.780385E−04	9.143568E−46	5.008138E−45
F11	0	0	1.064245E−05	8.705436E−06	1.332268E−16	5.701448E−16
F12	0	0	2.044922E−02	1.080113E−02	1.863431E+03	6.520088E+03
F13	0	0	5.264393E−09	7.659223E−09	6.96E−16	2.23E−15
F14	−1	0	N\A	N\A	0	0
F15	0	0	3.818312E−02	2.676411E−02	3.656060E−08	6.556170E−08
F16	−1	0	-9.996790E−01	2.197546E−04	−1	0
F17	0	0	1.360744E−04	8.183569E−05	2.664735E−02	8.517967E−02
F18	0	0	2.044922E−02	1.080113E−02	1.863431E+03	6.520088E+03
F19	0	0	3.666030E−12	3.523984E−12	6.480774E−125	1.449142E−124
F20	0	0	1.412541E−05	1.480976E−05	1.929200E−118	9.910730E−118
F21	0	0	2.282842E−06	1.836692E−06	5.553268E−17	3.041121E−16
F22	0	0	3.645288E−06	6.075621E−06	0	0

Function	GSA		GWO	
	Average	Std	Average	Std
F1	2.144713E−17	5.63657E−18	8.32056E−62	2.010429E−61
F2	9.310383E−11	5.16071E−11	8.7402E−216	0
F3	6.019471E−11	2.99343E−11	1.2596E−189	0
F4	−200	0	−200	0
F5	0	0	0	0
F6	1.852208E−20	1.43471E−20	1.540984E−07	1.113923E−07
F7	3.111632E−20	4.0912E−20	0	0
F8	1.569579E−21	1.18655E−21	8.417520E−204	0
F9	1.716123E−99	4.23303E−99	0	0
F10	1.522578E−10	6.4698E−11	3.205995E−214	0
F11	2.015060E−02	3.19937E−02	0	0
F12	2.839994E−02	3.32043E−02	1.068024E−01	2.613230E−01
F13	5.33E−03	6.96E−03	0	0
F14	3.39E−32	1.85E−31	1.56E−121	8.53E−121
F15	9.227673E−16	3.484625E−16	1.150598E−59	3.745600E−59
F16	−9.873900E−01	1.254131E−02	−9.957467E−01	1.618658E−02
F17	7.384492E−05	5.667909E−05	4.948649E−07	6.098266E−07
F18	2.839994E−02	3.320435E−02	1.068024E−01	2.613230E−01
F19	6.915319E−21	8.073127E−21	0	0
F20	1.280693E−20	1.039699E−20	0	0
F21	1.258348E−07	2.934570E−07	1.057100E−44	5.789975E−44
F22	1.148131E−02	3.443370E−02	0	0

Table 4: Results of multimodal benchmark functions

Function	RHSO		ABC		PSO	
	Average	Std	Average	Std	Average	Std
F23	8.88E−16	2.07883E−31	3.463270E−04	1.776345E−04	1.953996E−15	1.655893E−15
F24	4.22E−02	0.029420356	8.927490E−03	2.970026E−03	6.950133E−01	7.717305E−01
F25	−1.0316	2.34056E−16	−1.0316	6.775215E−16	1.575971E+01	9.146534E+01
F26	0.3979	0	0.39789	1.693804E−16	4.824710E−01	4.632838E−01
F27	0	0	2.562660E+01	4.714264E+00	1.401837E+01	2.396753E+01
F28	−3.7358	0.078818249	−3.8628	3.161767E−15	−3.862782	1.355043E−15
F29	0	0	2.496393E−09	2.058588E−09	2.27125E−100	8.66813E−100
F30	−195.629	2.99591E−14	−1.95629E+02	5.781517E−14	−1.95629E+02	3.566230E−05
F31	0	0	3.184262E−05	2.956543E−05	2.960595E−17	5.782380E−17
F32	−106.7645	0	−106.7645	7.226896E−14	−103.1151	1.998874E+01
F33	−2.06249565	0.000329985	−2.0626	1.355043E−15	−2.062339	1.496152E−03
F34	−0.6737	1.17028E−16	−0.67367	0	−0.6737	1.129203E−16
F35	−50	0	−49.9974	7.226896E−14	−50	0
F36	−0.869	1.36324E−06	−2.8739	9.033621E−16	−0.625	0
F37	1	0	1	0	1	0
F38	−1.8013	0	−1.878847	7.842049E−02	−1.801087	3.598212E−04
F39	0	0	8.444203E−01	1.298731E−01	6.81E−03	8.86E−03
F40	−1	0	3.291594E−10	1.683678E−10	1.25E−20	3.80E−20
F41	0	0	2.838598E+01	2.624794E+00	4.52E−04	8.97E−04
F42	0	0	3.562694E−05	3.280962E−05	0	0
F43	0	0	9.717305E−07	1.067286E−06	4.90E−123	1.66E−122
F44	0.9	0	9.019100E−01	2.128623E−03	9.17E−01	3.79E−02
F45	0	0	3.498900E−08	6.222377E−08	5.92E−11	1.04E−10
F46	−3.87E+03	0	−3.873700E+03	2.312607E−12	−3.87E+03	1.85E−12
F47	−3.80E−03	0.00E+00	−3.791200E−03	1.764379E−03	−3.80E−03	1.32E−18
F48	0	0	5.815390E−03	5.973223E−03	2.967771E−61	5.970806E−61

Function	GSA		GWO	
	Average	Std	Average	Std
F23	1.70832E−10	1.074615E−10	8.881800E−1	4.011733E−31
F24	4.47052E−03	2.057632E−03	9.716878E−04	9.025188E−04
F25	−1.0316	6.775215E−16	−1.03162	4.310840E−09
F26	0.3979	0	3.978879E−01	5.976276E−07
F27	3.117547E+00	1.325468E+00	0	0
F28	−3.862797	1.825742E−05	−3.862081	1.938044E−03
F29	6.25209E−21	7.707694E−21	0	0
F30	−195.629	5.781517E−14	−195.629	2.850603E−08
F31	0	0	0	0
F32	−106.7645	7.226896E−14	−1.061161E+02	3.551734
F33	−2.0626	1.355043E−15	−2.062612	3.302682E−09
F34	−0.6708967	4.172280E−03	−6.737000E−01	1.129203E−16
F35	−50	0	−4.999989E+01	8.030116E−05
F36	−0.869	1.129203E−16	−8.690000E−01	1.129203E−16
F37	1	0	1	0
F38	−1.8013	6.775215E−16	−1.8013	6.775215E−16
F39	3.71E+00	1.86E+00	3.379921E−03	7.16E−03
F40	3.94E−29	1.75E−29	2.87E−16	2.04E−16
F41	8.000533E−05	4.381770E−04	1.08E−04	2.70E−04
F42	0	0	0	0
F43	3.21E−19	2.98E−19	0	0
F44	9.18E−01	1.86E−02	9.13E−01	3.46E−02
F45	5.88E−06	9.54E−06	5.25E−12	9.11E−12
F46	−3.87E+03	1.85E−12	−3.87E+03	2.31E−12
F47	−3.80E−03	1.32E−18	−3.79E−03	1.76E−18
F48	2.089000E−02	1.933088E−02	0	0

For each benchmark function, the RHSO algorithm and the compared algorithms are performed in the experiments under the condition of the same number of iterations (1000), independent runs for 30 times, and the population size is set to 50. The statistical results (average and standard deviation) are showed in Tabs. 3 and 4. For verifying the results, the RHSO algorithm is compared with ABC [7], PSO [8], GSA [23], and GWO [4].

4.1 Exploitation Analysis

The results in Tab. 3 demonstrated that RHSO is better than the selected algorithms in all unimodal test functions. Unimodal functions test the exploitation of an algorithm. The obtained results showed RHSO superiority in exploiting the optimal value, so RHSO provides excellent exploitation ability.

4.2 Exploration Analysis

For testing the exploration strength of an algorithm, the multimodal functions are used as the number growing exponentially with dimension such types of functions. The results in Tab. 4 demonstrated that RHSO is better than the selected algorithms on most multimodal functions. The obtained results show the superiority of the RHSO algorithm in terms of exploration.

The multimodal results presented in Tab. 4 demonstrated the exploitation of the proposed algorithm. Those results show the efficiency and strength of RHSO as compared to the other algorithms that are adopted for comparison. RHSO is tested on 26 multimodal test functions, at which RHSO is better and more efficient than the other algorithms. Also, RHSO can control exploitation better than other algorithms.

Based on the results in Tab. 3, the RHSO algorithm obtained the optimal value in all test functions; Therefore, RHSO has surpassed the selected algorithms, and this makes the RHSO algorithm better than they do in exploitation convergence. Those results demonstrated the accuracy, efficiency, and flexibility of the proposed algorithm.

According to the results in Tab. 4, the RHSO algorithm obtained the optimal value in 23 test functions; also, it is very close to the optimal value and better than the values of the selected algorithms for the same test functions in three test functions (F23, F24, and F28).

5 Conclusions and Future Work

This paper proposed a new optimization algorithm inspired by Rock hyrax's behavior; RHSO is proposed as an alternative technique for solving optimization problems. In the proposed RHSO algorithm, the updating of position makes the solutions move towards or outwards the goal to guarantee the search space exploitation and exploration. Forty-eight test functions are used to test the performance of RHSO in terms of exploitation and exploration. The obtained results demonstrated that RHSO was able to outperform ABC, PSO, GSA, and GWO. The obtained unimodal test functions result demonstrated RHSO algorithm exploitation superiority. After that, RHSO exploration ability is shown by the obtained multimodal test functions result. RHSO algorithm is characterized by having few variables than the other algorithms, making it easy to understand and implement. RHSO can solve various optimization problems (like scheduling, maintenance, parameters optimization, and many others). Future studies can recommend several directions, such as solving various optimization problems and developing a multi-objective version of the RHSO algorithm as RHSO is currently a single objective optimization algorithm.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Muro, R. Escobedo, L. Spector and R. P. Coppinger, "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural Processes*, vol. 88, no. 3, pp. 192–197, 2011.

[2] J. J. L. Higdon and S. Corrsin, "Induced drag of a bird flock," *American Naturalist*, vol. 112, no. 986, pp. 727–744, 1978.

[3] E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. USA: Oxford University Press, 1999.

[4] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[5] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[6] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris *et al.,* "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[7] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *Proc. of the IEEE Swarm Intelligence Symp.*, Indianapolis, IN, USA, pp. 4–12, 2006.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of Int. Conf. on Neural Networks*, Perth, WA, Australia, pp. 1942–1948, 1995.

[9] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and Biological Systems: Towards a New Bionics?, NATO ASI Series (Series F: Computer and Systems Sciences)*, vol. 102. Berlin, Heidelberg, Germany: Springer, 1993.

[10] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm diversity," in *Proc. of IEEE World Congress on Computational Intelligence*, Hong Kong, pp. 1128–1134, 2008.

[11] S. Mirjalili, S. Zaiton and M. Hashim, "A new hybrid PSOGSA algorithm for function optimization," in *Proc. of Int. Conf. on Computer and Information Application*, Tianjin, pp. 374–377, 2010.

[12] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation," *Soft Computing*, vol. 13, no. 2, pp. 157–168, 2009.

[13] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.

[14] D. Goldberg, *Genetic Algorithms in Optimization, Search and Machine Learning*, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[15] K. P. Rainer Storn, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[16] L. Fogel, *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*, New York: John Wiley Sons, Inc., 1999.

[17] I. Rechenberg, "Evolution Strategy: Nature's way of optimization," in *Optimization: Methods and Applications, Possibilities and Limitations, Lecture Notes in Engineering*, vol. 47. Berlin, Heidelberg: Springer, pp. 106–126, 1989.

[18] N. Hansen, S. D. Müller and P. Koumoutsakos, "Reducing the time complexity of the randomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[19] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, no. 2, pp. 87–112, 1994.

[20] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[21] B. Webster and P. J. Bernhard, "A local search optimization algorithm based on natural principles of gravitation," in *Proc. of the Int. Conf. on Information and Knowledge Engineering*, Las Vegas, Nevada, USA, pp. 255–261, 2003.

[22] O. K. Erol and I. Eksin, "A new optimization method: Big bang–big crunch," *Information Sciences*, vol. 37, no. 2, pp. 106–111, 2006.

[23] E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, "GSA: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[24] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system," *Acta Mechanica*, vol. 213, no. 3–4, pp. 267–289, 2010.

[25] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13170–13180, 2011.

[26] A. Hatamlou, "Blackhole: A new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.

[27] M. K. A. Kaveh, "A new meta-heuristic method: Ray optimization," *Computers & Structures*, vol. 112, pp. 283–294, 2012.

[28] H. Z. J. Du and X. Wu, "Small-world optimization algorithm for function optimization," in *Proc. of the Second Int. Conf. on Advances in Natural Computation*, Xi'an, China, pp. 264–273, 2006.

[29] H. Shah-Hosseini, "Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimization," *International Journal of Computational Science and Engineering*, vol. 6, no. 1/2, pp. 132–140, 2011.

[30] F. F. Moghaddam, R. F. Moghaddam and M. Cheriet, "Curved space optimization: A random search based on general relativity theory," *ArXiv*, vol. abs/1208.2214, pp. 1–16, 2012.

[31] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[32] K. J. Brown, "Seasonal variation in the thermal biology of the rock hyrax (Procavia capensis)," MS thesis, Pietermaritzburg, School of Botany and Zoology, University of KwaZulu-Natal, 2003.

[33] K. J. Brown and C. T. Downs, "Basking behavior in the rock hyrax (Procavia capensis) during winter," *African Zool*, vol. 42, no. 1, pp. 70–79, 2007.

[34] L. Koren, O. Mokady and E. Geffen, "Social status and cortisol levels in singing rock hyraxes," *Hormones and Behavior*, vol. 54, no. 1, pp. 212–216, 2008.

[35] Rock hyrax, "San Diego Zoo Global Wildlife Conservancy," 2020. [Online]. Available: https://animals.sandiegozoo.org/animals/rock-hyrax.