Random-guided optimizer: a metaheuristic that shifts random search to guided search through iteration

Purba Daru Kusuma, Faisal Candrasyah Hasibuan
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received Apr 17, 2023
Revised Dec 25, 2023
Accepted Feb 24, 2024

Keywords:

Metaheuristic
Northern goshawk optimization
Optimization
Random motion
Slime mold algorithm
Stochastic process
Swarm intelligence

ABSTRACT

This study offers a new swarm-based metaheuristic: random-guided optimizer (RGO). RGO has novel mechanics in shifting the random motion into a guided motion strategy during the iteration. In RGO, the iteration is divided into three equal size phases. In the first phase, the unit walks randomly inside the search space to tackle the local optimal problem earlier. In the second phase, each unit uses a unit selected randomly among the population as a reference in conducting the guided motion. In the third phase, each unit conducts guided motion toward or surpasses the best unit. Through simulation, RGO successfully finds the acceptable solution for 23 benchmark functions. Moreover, RGO successfully finds the global optimal solution for four functions: Branin, Goldstein-Price, Six Hump Camel, and Schwefel 2.22. RGO also outperforms slime mold algorithm (SMA), pelican optimization algorithm (POA), golden search optimizer (GSO), and northern goshawk optimizer (NGO) in solving 12, 20, 12, and 1 function consecutively. In the future, improvement can be made by transforming RGO into solid multiple-phase strategy without losing its identity as a metaheuristic with multiple strategy in every iteration.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

Corresponding Author:

Purba Daru Kusuma
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University
Bandung, Indonesia
Email: purbodaru@telkomuniversity.ac.id

1. INTRODUCTION

Optimization is a famous and essential study. Its importance and popularity make optimization has been implemented in many areas, such as in computing system [1], transportation [2], mobile communication [3], production and manufacturing [4], healthcare [5], power system [6], home appliance [7], and so on. In production and manufacture, some study need optimization, such as in flow shop scheduling in multi factory environment [8], nozzle assignment in printed circuit board assembly [9], and job assignment in manufacture where there are several parallel machines [10], and so on. Several objectives in the manufacturing optimization problems are minimizing the completion time [8], total assembly time [9], make-span [11], maximizing service level [12], and so on. Meanwhile, the common objectives in the optimization work in the power system is higher energy harvesting [13], minimizing power loss [14], maximizing power point [15], and so on.

The metaheuristic has been used extensively in many optimization studies. This circumstance happens due to the advantage of metaheuristics in tackling complex optimization real-world problems, especially in the number of variables, constraints, or objectives [16]. This advantage comes from the characteristic of metaheuristic algorithm as approximate approach [17]. It makes the process does not trace all possible solutions that cost excessive computational resources [17].

A plenty of new metaheuristics were developed based on swarm intelligence. Many of them adopt foraging mechanics, such as slime mold algorithm (SMA) [18], grey wolf algorithm (GWO) [19], marine predator algorithm (MPA) [20], Komodo Mlipir algorithm (KMA) [21], pelican optimization algorithm (POA) [22], northern goshawk optimization (NGO) [23], butterfly optimization algorithm (BOA) [24], squirrel search optimizer (SSO) [25], tunicate swarm algorithm (TSA) [26] and so on. Several algorithms use the term leader as metaphors, such as three influential members-based optimizers (TIMBO) [27], mixed leader-based optimizer (MLBO) [28], and multileader optimizer (MLO) [29], hybrid leader-based optimization (HLBO) [30], and more.

The similarity among the swam-based metaheuristics is the existence of one or several units that become a reference for the swarm. These references can be the best unit, the worst unit, the randomly selected unit, and so on. A golden search optimizer (GSO) uses the best unit in every iteration and replaces the worst unit with the randomized unit among the population [31]. The population conducts the sinusoid motion toward the global and local best [31]. In HLBO, the reference combines the corresponding unit, the best unit, and the randomly selected unit [30]. Each unit's proportion is calculated based on the normalized fitness score [30]. Then, the corresponding unit walks toward the hybrid leader if this hybrid leader is better than the corresponding unit [30]. Otherwise, the corresponding unit walks away from this hybrid leader [30]. In MLBO, the mixed leader combines the best and randomly selected units in the first half of the iteration [28]. In the second half, the best unit is the only unit that constructs the mixed leader [28]. In GWO, the leader is constructed by the resultant of the three best units in every iteration [19]. In NGOs, the reference for every unit is selected randomly among the population [23]. Meanwhile, in POA, a reference is selected randomly from the problem space at the beginning of every iteration [22]. MLO is the more flexible version of GWO. MLO determines the number of best units selected in every iteration manually before the iteration runs [29]. In KMA, there are two types of leaders: the big males and the highest quality big male [21].

Based on this explanation, there are four issues usually exploited to construct a new swarm-based metaheuristic. The first is the construction of the reference. The second is the population's behavior in interacting with the reference. The third is randomized motion. The search space of the random motion can be the same in every iteration as in KMA [21], declines linearly as in MLO [29], or among the population. The fourth is the strategy controlled by the iteration. In most metaheuristic algorithms, the strategy chosen in every iteration is still the same. A few algorithms deploy different strategies along the iteration. In MPA, the iteration is split into three phases [20]. There is a distinct strategy deployed in every phase [20].

Despites the massive development of metaheuristics, the main problem is that there is no perfect metaheuristic as stated in the no-free-lunch theory [32]. For example, KMA is not the best one to solve Rosenbrock, Quartic, Penalized, and Penalized 2 [21]. BOA is not the best one to solve Ackley, Levy, Michalewiz, and Rosenbrock. GSO is not the best one to solve Step, Penalized, Penalized 2, Branin, and Hartman 6 [31]. The imperfection of metaheuristic creates room and space for the future development of metaheuristic. Although competition among metaheuristic has been criticized [32], exploring various method is still interesting. The secondary circumstance is the rarity of metaheuristics that changes their strategy during the iteration. It makes the development of metaheuristic with changing strategy during the iteration is challenging.

This study proposes a new metaheuristic algorithm: a random-guided optimizer (RGO). This algorithm deploys the iteration-controlled strategy where the iteration is split into several phases, and a distinct strategy is deployed in every phase. Based on its name, the strategy walks from random motion to guided motion. This mechanism is rare in the development of metaheuristic algorithms.

The scientific contribution of this work is presented:

a. A new swarm-based metaheuristic called RGO is proposed with its novel strategy is the shifting from full random to the guided search during iteration.
b. The performance of RGO is assessed by using the set of 23 functions as a theoretical use case.
c. The comparative performance of RGO, which consisting of its strength and weakness is benchmarked with four new swarm-based metaheuristics: SMA, POA, GSO, and NGO.

The outline of the following sections of this paper is formulated as follows. The research method, which consists of the presentation of the model and its assessment scenario, is presented in section 2. The simulation's findings, detailed examination of the results, insights from the simulation and the limitation of this work are investigated in section 3. The summarization of the conclusion and the possibility for further research is presented in section 4.

2. METHOD

2.1. Proposed model

This section presents the model of the proposed algorithm. The conceptual frame study of the algorithm, a formal description of the algorithm in pseudocode, and a mathematical model that formalizes the algorithm's construction process make up this presentation.

The term RGO comes from the basic concept where the optimization strategy shifts from random to guided motion during the iteration. The system consists of a certain number of units that represent the solutions. This set of units configures the population. The global best is the highest quality unit so far and has become the collective knowledge shared among the population. In RGO, the iteration is split into three equal size phases as in MPA [20]. In the first phase, every unit walks randomly inside the search space. This method is aimed to address the local optimum entrapment as soon as possible. In the second phase, each unit walks according to its local reference. This phase is known as a partial guided motion. This local reference is a unit that is selected randomly among the population. In the third phase, the corresponding unit walks toward or surpasses the global best. Each time a unit walks to a new one, the global best is updated in all phases. The illustration of this shifting is presented in Figure 1. This basic concept is then transformed into an algorithm. The algorithm of RGO is shown in Algorithm 1. Meanwhile, the annotations used in this study.

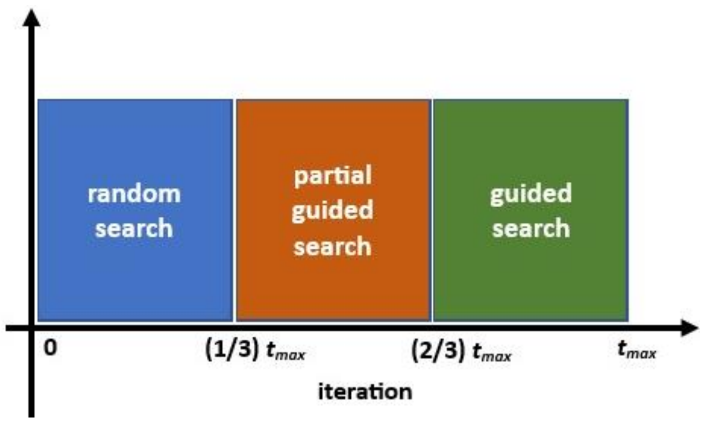

Figure 1. Shifting from random search to guided search

Algorithm 1. RGO

```
output: s_best
begin
  for i=1 to n(S) do
    generate the initial unit using (1)
    update s_best using (2)
  end for
  for t=1 to t_max  do
    for i=1 to n(S)do
      if t < t_max / 3 then
        conduct random motion using (1)
      else
        if t < 2t_max /3 then
          conduct partial guided motion using (3) to (5)
        else
          conduct guided motion using (6) and (5)
        end if
      end if
      update s_best using (2)
    end for
  end for
end
```

where:

b_l,	lower bound, upper bound	s_{sel}	selected unit
f	fitness function	s_{can}	unit candidate
s	unit	t	iteration
S	set of units	t_{max}	maximum iteration
s_{be}	best unit	U	uniform random

Here is the explanation of Algorithm 1. Line 1 shows that the global best becomes the output, i.e., the final unit. Initialization occurs between lines 3 and 6. The iteration phase is represented by lines 7 to 20. There are two loops inside the iteration. The out loop iterates from the first iteration to the maximum iteration. The inner loop iterates for the entire population. Lines 9 to 17 indicate the strategy chosen in every phase. Line 18 indicates that the global best is updated at the end of every motion.

The algorithm complexity of RGO can be presented as $O(t_{max}\,.\,n(S))$. It means that two variables affect the complexity linearly: the maximum iteration and the population size. Meanwhile, the mathematical model used in this algorithm is presented in (1) to (6):

$$s = U(b_l, b_u) \tag{1}$$

$$s_{best}{}' = \begin{cases} s, f(s) < f(s_{best}) \\ s_{best}, else \end{cases} \tag{2}$$

$$s_{sel} = U(S) \tag{3}$$

$$s_{can} = \begin{cases} s + 2U(0,1).(s_{sel} - s), f(s_{sel}) < f(s) \\ s_{sel} + 2U(0,1).(s - s_{sel}), else \end{cases} \tag{4}$$

$$s' = \begin{cases} s_{can}, f(s_{can}) < f(s) \\ s, else \end{cases} \tag{5}$$

$$s_{can} = s + 2U(0,1).(s_{best} - s) \tag{6}$$

Here is the explanation of (1) to (6). The initial unit is randomly selected from the search space and follows a uniform distribution, according to (1). In (2) states that the unit only replaces the optimal unit if it is superior to the optimal unit. According to (3), the selected unit is picked randomly from the population. In (4) states that the candidate is generated by moving the corresponding unit toward or surpassing the selected unit only if this selected unit is better than the corresponding unit. If not, the candidate is formed between the selected unit and the virtual unit that is distinct from the unit corresponding to the selected unit. In (5) states that the candidate replaces the current unit only if this candidate is better than the current unit. In (6) states that the candidate is generated by moving the corresponding unit toward and surpassing the global best.

2.2. Assessment scenario

RGO is then implemented into the simulation that tries to solve optimization problems. This study uses the well-known 23 benchmark functions as theoretical optimization problems. These functions represent all variants of theoretical single objective problems. Moreover, many studies proposing new metaheuristics use these functions for the theoretical use cases, such as in studies proposing KMA [21], GSO [31], and so on. They can be classified into three categories: (1) high dimensional unimodal functions (HDUF), (2) high dimensional multimodal functions (HDMF), and (3) fixed dimensional multimodal functions (FDMF). Functions 1 to 7 are HDUF functions. Functions 8 to 13 are HDMF functions. Functions14 to 23 are FDMF functions. These 23 functions also represent functions with various problem spaces, from the narrow ones, such as Hartman 3, Hartman 6, Rastrigin, or Quartic, to the wide ones, such as Griewank or Schwefel. A detailed description of these functions is presented in Table 1.

This study confronts RGO with four latest metaheuristic algorithms: SMA, POA, GSO, and NGO. All these algorithms use the swarm intelligence approach. SMA represents an algorithm that many optimization studies have used. SMA has been implemented and modified in several engineering problems, such as in optimizing the pressure vessel design [33], distribution system [34], conical peak cutting process [35], and hydropower multiple reservoir system [36]. On the other hand, POA, GSO, and NGO are brand-new algorithms first introduced in 2022. Meanwhile, later studies regarding these algorithms are still rare to find.

Table 1. 23 Benchmark functions

No	Function	Type	Dimension	Problem space	Global optimal
1	Sphere	HDUF	25	[-100, 100]	0
2	Schwefel 2.22	HDUF	25	[-100, 100]	0
3	Schwefel 1.2	HDUF	25	[-100, 100]	0
4	Schwefel 2.21	HDUF	25	[-100, 100]	0
5	Rosenbrock	HDUF	25	[-30, 30]	0
6	Step	HDUF	25	[-100, 100]	0
7	Quartic	HDUF	25	[-1.28, 1.28]	0
8	Schwefel	HDMF	25	[-500, 500]	-418.9×dim
9	Ratsrigin	HDMF	25	[-5.12, 5.12]	0
10	Ackley	HDMF	25	[-32, 32]	0
11	Griewank	HDMF	25	[-600, 600]	0
12	Penalized	HDMF	25	[-50, 50]	0
13	Penalized 2	HDMF	25	[-50, 50]	0
14	Shekel Foxholes	FDMF	2	[-65, 65]	1
15	Kowalik	FDMF	4	[-5, 5]	0.0003
16	Six Hump Camel	FDMF	2	[-5, 5]	-1.0316
17	Branin	FDMF	2	[-5, 5]	0.398
18	Goldstein-Price	FDMF	2	[-2, 2]	3
19	Hartman 3	FDMF	3	[1, 3]	-3.86
20	Hartman 6	FDMF	6	[0, 1]	-3.32
21	Shekel 5	FDMF	4	[0, 10]	-10.1532
22	Shekel 7	FDMF	4	[0, 10]	-10.4028
23	Shekel 10	FDMF	4	[0, 10]	-10.5363

3. RESULTS AND DISCUSSION

3.1. Assessment result

The result is presented in Tables 2 to 4. The average fitness score is shown in Table 2. The fitness score's standard deviation is displayed in Table 3. Table 4 displays the number of categories in which RGO outperforms its competitors. Several settings regarding this simulation are as follows. The maximum number of iterations is 100, which implies a low number of iterations. The population is set at 20, which is a reasonable number. Except for the maximum iteration and population size, none of the parameters of these algorithms can be modified.

Table 2. Simulation result (average fitness score)

F.	SMA	POA	GSO	NGO	RGO	Better than
1	$2.285x10^{3}$	$2.401x10^{4}$	$2.072x10^{3}$	$1.159x10^{-13}$	$5.424x10^{2}$	SMA, POA, GSO
2	0	0	$4.546x10^{28}$	0	0	GSO
3	$1.445x10^{4}$	$3.776x10^{4}$	$5.678x10^{3}$	$7.030x10^{-1}$	$2.132x10^{4}$	POA
4	$3.114x10^{1}$	$5.821x10^{1}$	$2.326x10^{1}$	$4.274x10^{-6}$	$5.311x10^{1}$	POA
5	$2.130x10^{6}$	$4.069x10^{7}$	$4.513x10^{5}$	$2.373x10^{1}$	$1.520x10^{5}$	SMA, POA, GSO
6	$2.816x10^{3}$	$2.275x10^{4}$	$1.837x10^{3}$	2.812	$5.329x10^{2}$	SMA, POA, GSO
7	$6.427x10^{1}$	$1.636x10^{1}$	$2.409x10^{-1}$	$3.425x10^{-3}$	1.510	SMA, POA
8	$-6.548x10^{3}$	$-3.172x10^{3}$	$-3.964x10^{3}$	$-4.228x10^{3}$	$-5.553x10^{3}$	POA, GSO, NGO
9	$2.759x10^{1}$	$2.703x10^{2}$	$8.982x10^{1}$	$1.855x10^{-6}$	$1.194x10^{2}$	POA
10	9.353	$1.920x10^{1}$	$1.879x10^{1}$	$9.693x10^{-8}$	$1.168x10^{1}$	POA, GSO
11	$2.127x10^{1}$	$2.026x10^{2}$	$1.759x10^{1}$	$4.167x10^{-9}$	5.627	SMA, POA, GSO
12	$3.916x10^{5}$	$5.667x10^{7}$	$1.191x10^{4}$	$2.880x10^{-1}$	$1.300x10^{3}$	SMA, POA, GSO
13	$3.959x10^{6}$	$1.051x10^{8}$	$5.128x10^{5}$	2.098	$3.321x10^{4}$	SMA, POA, GSO
14	1.359	2.206	5.463	1.171	1.800	POA, GSO
15	$9.169x10^{-2}$	$2.483x10^{-3}$	$3.006x10^{-3}$	$4.922x10^{-4}$	$2.569x10^{-3}$	SMA, GSO
16	$-1.439x10^{-1}$	-1.029	-1.032	-1.032	-1.032	SMA, POA
17	$6.345x10^{-1}$	$4.015x10^{-1}$	$3.981x10^{-1}$	$3.981x10^{-1}$	$3.981x10^{-1}$	SMA, POA
18	3.000	3.054	3.000	3.000	3.000	POA
19	$-4.954x10^{-2}$	$-4.954x10^{-2}$	$-4.954x10^{-2}$	$-4.954x10^{-2}$	$-4.954x10^{-2}$	-
20	-1.248	-3.025	-3.301	-3.303	-3.243	SMA, POA
21	-8.454	-4.108	-8.730	-8.843	-6.607	POA
22	-7.049	-3.781	-9.780	-9.648	-8.907	SMA, POA
23	-8.139	-3.230	-7.291	-9.885	-7.549	POA, GSO

Table 2 shows that RGO can find the acceptable solution for all 23 benchmark functions. RGO can find the sub-optimal solution of 19 functions. Meanwhile, RGO can find the optimal global solution of four functions: Schewfel 2.22, Six Hump Camel, Goldstein-Price, and Branin. Confronted with the sparing algorithms, RGO is better than three algorithms in solving seven functions, two algorithms in solving nine functions, and one algorithm in solving six functions. Meanwhile, all algorithms achieve the same result in solving Hartman 3. Based on this result, it can be said that RGO is competitive enough confronted with these algorithms. Meanwhile, Table 3 shows that the standard deviation of RGO in solving these 23 functions is moderate.

Table 3. Simulation result (standard deviation)

Function	SMA	POA	GSO	NGO	RGO
1	$1.180x10^{3}$	$5.023x10^{3}$	$1.004x10^{3}$	$1.304x10^{-13}$	$3.767x10^{2}$
2	0	0	$1.109x10^{29}$	0	0
3	$8.818x10^{3}$	$1.166x10^{4}$	$3.431x10^{3}$	1.088	$6.385x10^{3}$
4	9.443	6.514	3.070	$3.030x10^{-6}$	$1.359x10^{1}$
5	$2.064x10^{6}$	$1.705x10^{7}$	$4.956x10^{5}$	$9.298x10^{-2}$	$4.460x10^{5}$
6	$1.415x10^{3}$	$4.304x10^{3}$	$9.489x10^{2}$	$3.101x10^{-1}$	$4.287x10^{2}$
7	$3.348x10^{1}$	7.896	$6.500x10^{-2}$	$1.983x10^{-3}$	1.096
8	$4.644x10^{2}$	$4.064x10^{2}$	$6.872x10^{2}$	$3.126x10^{2}$	$7.780x10^{2}$
9	$1.027x10^{1}$	$2.155x10^{1}$	$2.444x10^{1}$	$7.943x10^{-6}$	$3.395x10^{1}$
10	1.882	$4.671x10^{-1}$	3.116	$1.319x10^{-7}$	2.003
11	$1.092x10^{1}$	$3.428x10^{1}$	6.074	$1.857x10^{-8}$	3.457
12	$4.691x10^{5}$	$2.212x10^{7}$	$4.448x10^{4}$	$6.044x10^{-2}$	$2.711x10^{3}$
13	$3.188x10^{6}$	$5.805x10^{7}$	$8.196x10^{5}$	$2.527x10^{-1}$	$1.197x10^{5}$
14	$4.894x10^{-1}$	1.966	3.354	$4.874x10^{-1}$	1.569
15	$4.131x10^{-2}$	$1.197x10^{-3}$	$5.750x10^{-2}$	$1.226x10^{-4}$	$5.620x10^{-3}$
16	$2.903x10^{-1}$	$2.263x10^{-3}$	$2.273x10^{-16}$	$2.560x10^{-10}$	$2.273x10^{-16}$
17	$4.250x10^{-2}$	$3.410x10^{-3}$	$1.136x10^{-16}$	$2.231x10^{-11}$	$1.136x10^{-16}$
18	0	$6.512x10^{-2}$	$4.345x10^{-11}$	$5.971x10^{-14}$	0
19	$2.128x10^{-17}$	$2.131x10^{-17}$	$2.131x10^{-17}$	$2.131x10^{-17}$	$2.131x10^{-17}$
20	$5.743x10^{-1}$	$1.202x10^{-1}$	$4.692x10^{-2}$	$1.918x10^{-2}$	$5.758x10^{-2}$
21	2.463	1.720	2.600	1.957	3.569
22	2.932	2.210	2.072	1.910	3.042
23	2.803	$7.945x10^{-1}$	3.332	1.496	3.549

Table 4. Supremacy comparison

Cluster	Number of functions outperformed by RGO			
	SMA	POA	GSO	NGO
1	4	6	4	0
2	3	6	5	1
3	5	8	3	0
Total	12	20	12	1

Table 4 shows that RGO is competitive when confronted with SMA, POA, and GSO but less competitive than NGO. RGO is very competitive with POA due to its achievement in outperforming POA in solving 20 functions. Meanwhile, RGO is competitive enough with SMA and GSO because it outperforms SMA and GSO in solving 12 functions. Unfortunately, RGO outperforms NGOs only in solving one function. The competitiveness of RGO to SMA, POA, and GSO occurs in all categories: high dimensional unimodal functions, high dimensional multimodal functions, and fixed dimensional multimodal functions. Meanwhile, the function where RGO outperforms GSO is a high-dimensional multimodal function.

3.2. Discussion

In general, the simulation result shows that the strategy implemented in RGO performs well in making RGO a good metaheuristic algorithm. The result also shows that RGO is competitive with other latest metaheuristic algorithms, especially those built based on swarm intelligence. Table 4 shows that RGO can tackle both unimodal and multimodal functions. The random motion strategy conducted in the first phase allows RGO to avoid the local optimal entrapment. On the other hand, partial guided and guided motion conducted in the second and third phases makes RGO find the acceptable solution once it can find the area of the global optimal solution.

The simulation result strengthens the no-free-lunch theory because RGO is good at solving some functions but not so good at solving other functions. Although RGO outperforms POA, POA outperforms RGO in solving Kowalik. On the other hand, although, in general, RGO is inferior to NGO, RGO can outperform NGO in solving Schwefel.

The superiority of NGO confronted with RGO can be traced back to the number of sequential motions performed in every iteration. There are two sequential motions performed in NGO. The first one is guided motion while the second one is random motion [23]. It is different from RGO that performs single motion in every iteration. This result underlines that multiple and sequential motions in every iteration promises potential superiority. The superiority of NGO compared to RGO indicates the necessity of the interaction with other units. NGO implements this strategy along the iteration. Meanwhile, RGO implements this strategy only in one third of the iteration.

Meanwhile, the superiority of RGO over POA also indicates the necessity or existence of the global best unit as the reference. Although POA performs multiple and sequential motions in every iteration [22], the missing of the best unit as the reference makes POA inferior confronted with RGO. POA depends only on the randomized unit as a reference [22].

The superiority of RGO over GSO and SMA indicates the necessity of the rigid acceptance rule. RGO, NGO, and POA are metaheuristics with rigid acceptance rule so that the candidate should be better than the current solution for replacement to take place. On the other hand, both GSO and SMA do not implement this rule so that worse candidate still can replace the current solution.

There are several limitations regarding this study, especially the proposed RGO. First, RGO is still inferior to NGO although RGO is proven superior to POA, SMA, and GSO. Second, RGO has not been challenged to solve any real-world optimization problem, especially numerical optimization. It means that the success of RGO is limited to the theoretical problem. Third, in the primary form of RGO, the iteration is split into three equal size phases. Fourth, RGO still can be seen as a single-phase metaheuristic although it performs three strategies.

This limitation can be explored as a baseline for future development. Technically, there is an opportunity to modify the phase width to become more flexible. Moreover, this phase split strategy can be conducted more adaptively. In the primary form of RGO, the maximum step size is twice the distance between the corresponding unit and its reference. This strategy can be modified by widening or narrowing this maximum step size, and then the performance can be evaluated. In RGO, the uniform distribution is conducted in the stochastic process. So, the modification of the stochastic distribution with normal, exponential, or other distribution can be chosen. Technical improvement also can be made by deploying the iteration-controlled local search as found in NGO and POA, especially to make RGO more competitive compared to NGO. Moreover, the single-phase approach in the original form or RGO can be transformed into multiple-phase approach by adding more searching methods without losing the fundamental essence of RGO as a metaheuristic that implements iteration-controlled strategy. Meanwhile, it is important to utilize RGO to solve many optimization problems with real-world use cases.

4. CONCLUSION

This study has presented a new swarm intelligence-based metaheuristic algorithm, namely a RGO. The RGO has been evaluated to find the optimal unit for 23 benchmark functions. Through simulation, it is shown that RGO successfully finds the acceptable, i.e., sub-optimal unit of these 23 functions. Moreover, RGO successfully finds the optimal global unit of four functions: Schwefel 2.22, Branin, Goldstein-Price, and Six Hump Camel. RGO is also competitive when confronted with the confronting algorithms. RGO outperforms SMA, POA, GSO, and NGO in 12, 20, 12, and 1 function consecutively. Through comparative assessment, it is found that the existence of global best unit as reference, implementation of rigid acceptance rule, and multiple-phase strategy is important to make a powerful metaheuristic.

Several future studies can be conducted regarding this study. Although RGO is proven good in solving theoretical optimization problems, it should be implemented to solve various real-world optimization problems, from engineering to finance. Implementing RGO in solving integer-based optimization problems is also essential, as in many operations research studies. Transforming the RGO to solve the combinatorial optimization problem is also challenging. Moreover, technical improvement can be made by transforming RGO into a multiple-phase metaheuristics without losing its fundamental essence as a metaheuristic that uses iteration to determine the strategy it takes.

ACKNOWLEDGEMENTS

The authors acknowledge Telkom University for supporting the publication of this paper.

REFERENCES

[1] H. Wang, L. Cai, X. Hao, J. Ren, and Y. Ma, “ETS-TEE: An energy-efficient task scheduling strategy in a mobile trusted computing environment,” *Tsinghua Science and Technology*, vol. 28, no. 1, pp. 105–116, Feb. 2023, doi: 10.26599/TST.2021.9010088.

[2] Y. Wang, Q. Luo, and Y. Zhou, "Improved grey wolf optimizer for multiple unmanned aerial vehicles task allocation," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, pp. 577–585, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp577-585.
[3] A. Bechihi, E. Panteley, P. Duhamel, and A. Bouttier, "A resource allocation algorithm for formation control of connected vehicles," *IEEE Control Systems Letters*, vol. 7, pp. 307–312, 2023, doi: 10.1109/LCSYS.2022.3187824.
[4] G. Ni and L. Chen, "Improved scheduling for the three-machine proportionate open shop and mixed shop minimum makespan problems," *IEEE Access*, vol. 8, pp. 186805–186812, 2020, doi: 10.1109/ACCESS.2020.3030694.
[5] R. Mothkur and V. B. Nagendrappa, "An optimal model for classification of lung cancer using grey wolf optimizer and deep hybrid learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, pp. 406–413, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp406-413.
[6] M. Zadehbagheri, T. Sutikno, M. J. Kiani, and M. Yousefi, "Designing a power system stabilizer using a hybrid algorithm by genetics and bacteria for the multi-machine power system," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1318–1331, Jun. 2023, doi: 10.11591/eei.v12i3.4704.
[7] B. H. Jasim, A. M. AL-Aaragee, A. A. A. Alawsi, and A. M. Dakhil, "A heuristic optimization approach for the scheduling home appliances," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1256–1266, Jun. 2023, doi: 10.11591/eei.v12i3.3989.
[8] Y. Yang and X. Li, "A knowledge-driven constructive heuristic algorithm for the distributed assembly blocking flow shop scheduling problem," *Expert Systems with Applications*, vol. 202, p. 117269, Sep. 2022, doi: 10.1016/j.eswa.2022.117269.
[9] H.-P. Hsu, "Solving the feeder assignment, component sequencing, and nozzle assignment problems for a multi-head gantry SMT machine using improved firefly algorithm and dynamic programming," *Advanced Engineering Informatics*, vol. 52, p. 101583, Apr. 2022, doi: 10.1016/j.aei.2022.101583.
[10] D. R. Page, R. Solis-Oba, and M. Maack, "Makespan minimization on unrelated parallel machines with simple job-intersection structure and bounded job assignments," *Theoretical Computer Science*, vol. 809, pp. 204–217, Feb. 2020, doi: 10.1016/j.tcs.2019.12.009.
[11] B. Xi and D. Lei, "Q-learning-based teaching-learning optimization for distributed two-stage hybrid flow shop scheduling with fuzzy processing time," *Complex System Modeling and Simulation*, vol. 2, no. 2, pp. 113–129, Jun. 2022, doi: 10.23919/CSMS.2022.0002.
[12] W. Cao and X. Wang, "A multiobjective multiperiod mixed-integer programming optimization model for integrated scheduling of supply chain under demand uncertainty," *IEEE Access*, vol. 10, pp. 63958–63970, 2022, doi: 10.1109/ACCESS.2022.3183281.
[13] N. A. Windarko *et al.*, "Hybrid photovoltaic maximum power point tracking of Seagull optimizer and modified perturb and observe for complex partial shading," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 5, pp. 4571–4585, Oct. 2022, doi: 10.11591/ijece.v12i5.pp4571-4585.
[14] J. A. O-. Toro, O. D. Montoya, and L. F. G-. Noreña, "Recursive convex approximations for optimal power flow solution in direct current networks," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, pp. 5674–5682, Dec. 2022, doi: 10.11591/ijece.v12i6.pp5674-5682.
[15] M. F. Alvianandy, N. A. Windarko, and B. Sumantri, "Maximum power point tracking based on improved spotted hyena optimizer for solar photovoltaic," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, pp. 5775–5788, Dec. 2022, doi: 10.11591/ijece.v12i6.pp5775-5788.
[16] J. Del Ser *et al.*, "Bio-inspired computation: Where we stand and what's next," *Swarm and Evolutionary Computation*, vol. 48, pp. 220–250, Aug. 2019, doi: 10.1016/j.swevo.2019.04.008.
[17] H. R. Moshtaghi, A. T. Eshlaghy, and M. R. Motadel, "A comprehensive review on meta-heuristic algorithms and their classification with novel approach," *Journal of Applied Research on Industrial Engineering*, vol. 8, no. 1, pp. 63–89, 2021, doi: 10.22105/jarie.2021.238926.1180.
[18] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.
[19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[20] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, p. 113377, Aug. 2020, doi: 10.1016/j.eswa.2020.113377.
[21] S. Suyanto, A. A. Ariyanto, and A. F. Ariyanto, "Komodo mlipir algorithm," *Applied Soft Computing*, vol. 114, pp. 1–17, Jan. 2022, doi: 10.1016/j.asoc.2021.108043.
[22] P. Trojovský and M. Dehghani, "Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications," *Sensors*, vol. 22, no. 3, pp. 1–34, Jan. 2022, doi: 10.3390/s22030855.
[23] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems," *IEEE Access*, vol. 9, pp. 162059–162080, 2021, doi: 10.1109/ACCESS.2021.3133286.
[24] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, Feb. 2019, doi: 10.1007/s00500-018-3102-4.
[25] M. Suman, V. Sakthivel, and P. Sathya, "Squirrel search optimizer: nature inspired metaheuristic strategy for solving disparate economic dispatch problems," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 111–121, Oct. 2020, doi: 10.22266/ijies2020.1031.11.
[26] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, pp. 1–30, Apr. 2020, doi: 10.1016/j.engappai.2020.103541.
[27] F. Zeidabadi, M. Dehghani, and O. Malik, "TIMBO: three influential members based optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 121–128, Oct. 2021, doi: 10.22266/ijies2021.1031.12.
[28] F. Zeidabadi, S. Doumari, M. Dehghani, and O. P. Malik, "MLBO: mixed leader based optimizer for solving optimization problems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 472–479, Aug. 2021, doi: 10.22266/ijies2021.0831.41.
[29] M. Dehghani *et al.*, "MLO: multi leader optimizer," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 364–373, Dec. 2020, doi: 10.22266/ijies2020.1231.32.
[30] M. Dehghani and P. Trojovský, "Hybrid leader based optimization: a new stochastic optimization algorithm for solving optimization applications," *Scientific Reports*, vol. 12, no. 1, pp. 1–16, Dec. 2022, doi: 10.1038/s41598-022-09514-0.
[31] M. Noroozi, H. Mohammadi, E. Efatinasab, A. Lashgari, M. Eslami, and B. Khan, "golden search optimization algorithm," *IEEE Access*, vol. 10, pp. 37515–37532, 2022, doi: 10.1109/ACCESS.2022.3162853.

[32] J. Swan *et al.*, “Metaheuristics “In the Large”,” *European Journal of Operational Research*, vol. 297, no. 2, pp. 393–406, Mar. 2022, doi: 10.1016/j.ejor.2021.05.042.
[33] B. N. Örnek, S. B. Aydemir, T. Düzenli, and B. Özak, “A novel version of slime mould algorithm for global optimization and real world engineering problems: Enhanced slime mould algorithm,” *Mathematics and Computers in Simulation*, vol. 198, pp. 253–288, Aug. 2022, doi: 10.1016/j.matcom.2022.02.030.
[34] Md. S. Abid, H. J. Apon, A. Ahmed, and K. A. Morshed, “Chaotic slime mould optimization algorithm for optimal load-shedding in distribution system,” *Ain Shams Engineering Journal*, vol. 13, no. 4, p. 101659, Jun. 2022, doi: 10.1016/j.asej.2021.101659.
[35] J. Zhou, Y. Dai, K. Du, M. Khandelwal, C. Li, and Y. Qiu, “COSMA-RF: New intelligent model based on chaos optimized slime mould algorithm and random forest for estimating the peak cutting force of conical picks,” *Transportation Geotechnics*, vol. 36, p. 100806, Sep. 2022, doi: 10.1016/j.trgeo.2022.100806.
[36] I. Ahmadianfar *et al.*, “Multi-strategy slime mould algorithm for hydropower multi-reservoir systems optimization,” *Knowledge-Based Systems*, vol. 250, p. 109048, Aug. 2022, doi: 10.1016/j.knosys.2022.109048.

BIOGRAPHIES OF AUTHORS

Purba Daru Kusuma received his Bachelor, and Master’s degrees in Electrical Engineering from Bandung Institute of Technology, Indonesia, in 2002 and 2009 respectively. He received his Doctoral degree in Computer Science from Gadjah Mada University, Indonesia, in 2017. Currently, he is an assistant professor in Telkom University, Indonesia. His research area is artificial intelligence, machine learning, and operational research. He can be contacted by email: purbodaru@telkomuniversity.ac.id.

Faisal Candrasyah Hasibuan he is a Computer Engineering lecturer at Telkom University, Bandung, West Java, Indonesia. Status as an almamater, he completed a bachelor's program at the place where he studys now. He was granted a Master’s degree in Electrical Engineering with a Computer Engineering specialization at Bandung Institute of Technology (ITB). The field of research interests are but are not limited to the embedded system, internet of things, and intelligent control system. He can be contacted by email: faicanhasfcb@telkomuniversity.ac.id.