An Approach Inspired from Nuclear Reaction Processes for Numerical Optimization

WEI Zheng-lei\textsuperscript{a} ZHANG Zhuo-ran\textsuperscript{b} HUANG Chang-qiang\textsuperscript{c} HAN Bo\textsuperscript{d} TANG Shang-qin\textsuperscript{e} WANG Le\textsuperscript{f}

Affiliation: Air Force Engineering University, Address: No.1, Baling road, Baqiao district, Xi’an, Shaanxi province

\textsuperscript{a}Zhenglei_wei@126.com
\textsuperscript{b}Zhuoran1009@163.com
\textsuperscript{c}cnxahcq@126.com
\textsuperscript{d}295629469@qq.com
\textsuperscript{e}carnationtang2@163.com
\textsuperscript{f}542127664@qq.com

ABSTRACT: As optimization problems become more complex, the need of new algorithms becomes greater. In this paper, a new powerful algorithm based on nuclear reaction process, namely, nuclear reaction optimization, is proposed for numerical optimization. The NRO algorithm which includes nuclear fission and fusion phases mimics the fission and fusion processes. In the first phase, the Gaussian walk and differential evolution operators are used for modeling fission process. In the second phase, the variants of differential evolution operators are employed for modeling the ionization stage and fusion stage. Additionally, six benchmark functions and unconstrained engineering design problem are evaluated by NRO and other compared algorithms. The optimization results indicate that the NRO has presented feasibility and superiority performance.

1. INTRODUCTION
Optimization algorithms play an important role in the fields of industry, agriculture, national defense, transportation, economy and management. Due to the deficiency that an initial guess can impact upon the solution quality, the traditional gradient-based optimization algorithms such as nonlinear programming, dynamic programming, sequential quadratic programming have not effectively managed nonconvex, disconnected and high-dimensional problems. Based on above analysis, many stochastic algorithms which are inspired from nature system have been proposed to solve these types of NP-hard problems [1]. In stochastic algorithms, some possible solutions are initially selected, then during the iterative process, this population searches for the solution domain without substantial gradient information until a well-pleasing solution is found or the number of iterations reaches the maximum.

These stochastic algorithms inspired from nature system can be categorized as evolution algorithms (EAs), swarm intelligence algorithms (SIAs), physics-inspired algorithms (PIAs) and human behavior-based algorithms (HBAs) [2][3]. Differential evolution (DE) [4] and genetic algorithm (GA) [5] are two typical representatives of EAs. GA is inspired from Darwin’s theory of biological evolution. Similar to GA, DE employs the mutation, cross and selection operators to find the optimum value. DE is proved to be more efficiency than GA, thus more variants of DE have been proposed. SIAs are in vigorous development and originated from the clustered behaviors in nature. The most representative algorithms
inspired creatures colonies are particle swarm optimization (PSO) [6], ant colony algorithm (ACO) [7], artificial bee colony algorithm (ABC) [8], bat algorithm (BA) [9] and cuckoo search (CS) [10]. In recent years, most state-of-the-art SIAs are proposed, such as grey wolf optimizer (GWO) [11], grasshopper optimization algorithm (GOA) [12], slap swarm algorithm (SSA) [13], butterfly-inspired algorithm [14], whale optimization algorithm (WOA) [15], squirrel search algorithm [16], symbiotic organisms search (SOS) [17] and spider monkey optimization (SMO) [18]. For PIAs, they are inspired by physical and chemical systems. The novel approaches include Gravitational search algorithm (GSA) [19], Chemical reaction optimization (CRO) [20], lightning search algorithm (LSA) [21] and atom search optimization (ASO) [22]. The inspirations of HBAs are social behavior of human beings. Recent literature has proposed quantities of popular HBAs, which include teaching-learning based optimization (TLBO) [23], cognitive behavior optimization algorithm (COA) [24] and human mental search (HMS) [25].

Although various novel algorithms spring up like mushrooms after raining, the No Free Lunch (NFL) theorem has determinately proved that there is no single algorithm that can suit best for all optimization problems. The NFL theorem motivates searchers in this field to present novel algorithms. Meanwhile, this is a very vital season why to develop a novel meta-heuristic algorithm inspired from nuclear reaction.

In our work, a novel and powerful meta-heuristic optimization algorithm, nuclear reaction optimization (NRO), is proposed. In NRO, it can be assumed that two crucial phases which include nuclear-fission (NFi) phase and nuclear-fusion (NFu) phase are performed well and modeled in a search space. NFi phase inspired from the process of heavy nucleus fission consists of three states for fragments of bombardment. According to nucleus type and decay state, the Gaussian walk and variants of differential evolutions operators are employed for imitating the fission process. NFu phase inspired from the process of light nuclei fusion represents as exploitation. The variants of differential evolutions operators and Levy flight strategies are employed for simulating the ionization and fusion stages. Two types of tests are performed, and the results prove the efficiency of our proposed algorithm due to a balance of exploration and exploitation.

The reminder of this paper is organized as follows: The proposed NRO algorithm is detailed in Section 2. Section 3 presents the statistical results of experiments. Finally, Section 4 gives the main conclusions of this work and offers directions for future study.

2. PROPOSED NUCLEAR REACTION OPTIMIZATION

2.1 A review of nuclear reaction
The proposed method has employed both nuclear fission and nuclear fusion theories. Minutely speaking, NRO employs several strategic and simple rules to search optimal solution. Firstly, in the nuclear fission process, the nuclei fragments after absorbing heated neutrons become odd nuclei and even-even nuclei. The odd nuclei can be divided into primary fission products which can be employed for exploitation and secondary fission products that can be used for exploration. The even-even nuclei which aren’t in the fission can search in neighborhood of current positions. Secondly, each nucleus is heated to be in the plasma state through absorbing the energy from nuclear fissions. Some nuclei which overcome Coulomb repulsion force and can be bound together by strong nuclear force can play a role of exploration. Other nuclei which are restrained by Coulomb repulsion force reduce the approaching velocity for exploitation or repel each other for exploration. The movement energy of each nuclei stems from the heated neutron or energy released in nuclear reaction.

2.2 Nuclear reaction optimization
In order to balance the abilities of exploration and exploitation, NRO assumes that the circulation in which NFi phase functions firstly and NFu phase is activated then could be feasible. Each item of a nucleus characteristic which includes some items, such as mass number of nucleus, charge property and number of atom and position represents each variable of a solution. Every nucleus is estimated by the
specific binding energy which is the binding energy per mass that represents the stability of nucleus, and the most stable nucleus is the optimal solution. The specific algorithm is detailed as follows:

2.2.1 Initialization of nuclei population
The initialization of the $i$th nucleus in the population is done by following formula:

$$X_i = l_b + \text{rand} \cdot (u_b - l_b)$$

(1)

where $X_i$ is the state of the $i$th nucleus of the population in a confined volume, $l_b$ and $u_b$ represent the lower and upper bounds of the $d$th variable in the search space respectively, and $\text{rand}$ is a uniformly distributed random number between 0 and 1.

2.2.2 NF$i$ phase
In the NF$i$ phase, the heated neutrons can be generated by mean of two different random nuclei. In generally, there are three states, which consist of odd nuclei fission state with producing secondary fission products after $\beta$ decay, odd nuclei fission state with primary fission products and even-even nuclei fission state shown as follows:

$$X_i^s = \begin{cases} \text{Gaussian} (X_{we}, \sigma_i) + (\text{randn} \cdot X_{we} - P_{ne}) & \text{rand} \leq P_i \& \text{rand} \leq P_e \\ \text{Gaussian} (X_i, \sigma_i) + (\text{rand} \cdot X_i - P_{ne}) & \text{rand} \leq P_i \& \text{rand} > P_e \\ \text{Gaussian} (X_i, \sigma_i) & \text{rand} > P_i \\ \end{cases}$$

(2)

where $X_i^s$ is the $i$th fission product nucleus, $\text{randn}$ is a normally distributed random number between 0 and 1, $X_{we}$ is the current best nucleus, $\sigma_i$ is $\{\log(g)/g\} \cdot (X_i - X_{we})$ where $g$ is the current generation, $\sigma_r$ is $\{\log(g)/g\} \cdot (X_i - X_{we})$, $P_{ne}$ is the heated neutrons. Additionally, for the first row of Eq. (2), the item $P_{ne} \in [1,2]$ means that SF product can exploit the smaller range of searching, where $\text{round}$ is nearest integer and $\text{rand}$ is uniformly distributed random number between 0 and 1. For the second row of Eq. (2), the item $P_{ne} \in [2,3]$ means that PF product can exploit the wider range of searching. For the last row of Eq. (2), the formula means the information of current nucleus could greatly be retained by Gaussian walk.

2.2.3 NF$u$ phase
In general, this phase is divided into two stages: ionization stage and fusion stage. For the first stage, differential evolution operators can be employed for ionization. For the second stage, fusing, approaching and repelling of the ions is modeled by variants of differential evolution operator.

For ionization stage, at first, all nuclei are ranked according to the fitness value. The ionization stage can be depicted as follows:

$$X_{i,j}^{\max} = \begin{cases} X_{i,j}^r + \text{rand} \cdot (X_{i,j}^r - X_{i,j}^u) & \text{rand} > P_a \& \text{rand} \leq 0.5 \\ X_{i,j}^r - \text{rand} \cdot (X_{i,j}^r - X_{i,j}^u) & \text{rand} > P_a \& \text{rand} > 0.5 \\ X_{i,j}^r + \text{round} \cdot \text{rand} \cdot (X_{i,j} - X_{i,j}^u) & \text{rand} \leq P_a \\ \end{cases}$$

(3)

where $X_{i,j}^{\max}$ represents the $d$th variable of $i$th ion after ionization, and $X_{i,j}^r$, $X_{i,j}^u$ and $X_{i,j}^d$ represent the $d$th variables of $i$th, $r$th and $d$th fission nuclei respectively. Additionally, $X_{\text{round},d}$ and $X_{\text{best},d}$ is the $d$th worst fission product nucleus. $P_a$ represents the probability of ionization.

Before the fusion stage, all ions obtained from ionization are ranked. The fusion stage can be depicted as follows:

$$X_{i,j}^{\max} = \begin{cases} X_{i,j}^r + \text{rand} \cdot (X_{i,j}^r - X_{i,j}^u) + \text{rand} \cdot (X_{i,j}^r - X_{i,j}^u) & \text{rand} > P_e \\ e^{-\alpha \cdot G_{\text{max}} \cdot (X_{i,j}^r - X_{i,j}^u)} \cdot \left( X_{i,j}^r - X_{i,j}^u \right) & \text{rand} \leq P_e \& \text{rand} \leq 0.5 \\ X_{i,j}^{\max} - 0.5 \cdot \sin (2 \pi \cdot \text{freq} \cdot g + x) \cdot \frac{G_{\text{max}} - 1}{G_{\text{sum}}} \cdot \left( X_{i,j}^r - X_{i,j}^u \right) & \text{rand} \leq P_e \& \text{rand} > 0.5 \\ X_{i,j}^{\max} - 0.5 \cdot \sin (2 \pi \cdot \text{freq} \cdot g + x) \cdot \frac{1}{G_{\text{sum}}} \cdot \left( X_{i,j}^r - X_{i,j}^u \right) & \text{rand} \leq P_e \& \text{rand} \leq 0.5 \\ \end{cases}$$

(4)

$$...
where $X_{i}^{f_{w}}$ is the $i$th fusion product, $X_{r1}^{f_{w}}$ and $X_{r2}^{f_{w}}$ represent respectively the $r_{1}$th and $r_{2}$th ions in which $r_{1}$ and $r_{2}$ are different. Additionally, $freq$ represents frequency of sinusoidal function.

In the NFu phase, when the difference item is zero, the Levy Flight strategies are employed which can be referred in literature [26]. In each phase, the nucleus with best fitness value in current iteration must be stored. The individuals beyond search boundary are recreated by using the boundary control strategy [26].

3. NUMERICAL EXPERIMENTS

In this section, in order to illustrate the feasibility and effectiveness of the proposed NRO, six standard benchmarks and a constrained engineering design problem are employed. These experiments are performed on a computer with a Core(TM) i7-7700HQ 2.80GHz processor with 8 GB RAM and Matlab 2015a is employed for the programming.

3.1 Experiment results on standard benchmarks

These six benchmarks functions, i.e., Sphere, Rosenbrock, Step, Rastrigin, Ackley and Penalized2, have different characteristics, which can be referred in [26]. Sphere, Rosenbrock and Step are designed to test the convergence rates. Meanwhile, the last three are designed to test the accuracy of optimization. Moreover, the proposed NRO is compared with GWO, WOA, CS and HS. The population size of each algorithm is set to 100. In order to make a fair comparison, the population size of NRO and CS which have two phases is set to 50. Meanwhile, the maximum function evaluations (MaxFEs) are set to 500 000 for the six benchmarks. For each benchmark, all compared algorithms are executed independently 30 times.

The statistical results which include mean solution and standard deviation derived by these five compared techniques are given in Table 1. The bold data obtained from the best solution are presented according to the mean values. As you can see in Table 1, proposed NRO shows the feasibility and superiority. Additionally, in Rastrigin, GWO possesses same results as NRO. However, in general, NRO performs more efficient than other compared algorithm in this test.

Table 1. Comparison of the statistical results of 5 algorithms for 6 standard benchmarks

| No. | GWO Mean/SD | WOA Mean/SD | CS Mean/SD | HS Mean/SD | NRO Mean/SD |
|-----|-------------|-------------|------------|------------|-------------|
| 1   | 4.78e-150/ 2.50e-73/ | 3.73e-09/ | 1.36e-02/ | 1.80e-177/ |             |
| 2   | 1.87e-149 1.13e-72 | 1.17e-10 | 1.31e-02 | 0.00e+00   |             |
| 3   | 2.57e+01/ 2.82e+01/ | 1.97e+01/ | 9.85e+01/ | 2.71e-23/  |             |
| 4   | 7.66e-01 2.73e-01 | 2.73e+00 | 3.57e+01 | 3.05e-23   |             |
| 5   | 7.49e-02/ 4.04e-01/ | 4.08e-10/ | 1.82e-02/ | 0.00e+00   |             |
| 6   | 1.16e-01 2.22e-01 | 1.16e-10 | 1.41e-02 | 0.00e+00   |             |

3.2 Application of NRO in solving constrained engineering design problem

In order to test the performance of NRO in searching optimal value for engineering design optimization problem, a constrained engineering design problem, namely, cantilever beam design problem, are test. In order to solve the constrained item, the penalty function method is applied. Meanwhile, this problem is independently carried out by NRO with 50 nuclei 30 times. MaxFEs is set to 500 000 for this problem during each optimization process.
This engineering design problem firstly defined by Chickermane and Gea, which is often employed for a constrained benchmark problem, consists of five elements shown in Figure 1. In the problem, each of the hollow cross section has a fixed diameter. The support of beam is rigid as shown, and at the free end of the cantilever beam, a vertical force is applied. The aim of the problem is to minimize the beam’s weight. The design variables represent the heights (or widths) of five beam elements. The problem is presented as follows:

\[
\min f(x_1, x_2, x_3, x_4, x_5) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)
\]  

Subject to:

\[
g(X) = \frac{61}{x_1} + \frac{37}{x_2} + \frac{19}{x_3} + \frac{7}{x_4} + \frac{1}{x_5} - 1 \leq 0
\]

Where the variables satisfy \(0.01 \leq x_i \leq 100\).

The statistical results obtained from all compared algorithms which include GWO, SOS, COA, CoBiDE and NRO are displayed in Table 2. As you can see in Table 2, the best value is 13.03204261 with 58 200 FEs obtained by NRO. Although COA outperforms in term of the number of function evaluations, NRO obtains the best fitness value. Thus, NRO could offer a competitive best fitness value with less number of FEs.

Table 2 Statistical results of different algorithms for Cantilever beam design problem.

|       | GWO       | SOS      | COA       | CoBiDE    | NRO       |
|-------|-----------|----------|-----------|-----------|-----------|
| Best  | 13.03208839 | 13.03206793 | 13.03204278 | 13.03558056 | **13.03204261** |
| Worst | 13.03291421 | 13.03242889 | 13.03208571 | 13.05340115 | **13.03204261** |
| Mean  | 13.03243341 | 13.03218901 | 13.03204994 | 13.04266822 | **13.03204261** |
| SD    | 2.3313e-04  | 8.6172e-05  | 1.0176e-05  | 4.7049e-03  | **9.0336e-15**  |
| MaxFEs| 490 000    | 66 300    | **56 400** | 94 200    | 58 200    |

4. CONCLUSIONS

According to NFL theorem and the inspirations of nuclear reaction, a novel meta-heuristic algorithm (called NRO, nuclear reaction optimization) which is inspired by nuclear fission and fusion processes is presented. NRO consists of two phases, which include NFi phase and NFu phase. In first phase, the Gaussian walk and differential evolution operator are employed for exploitation. In second phase, the variants of differential evolution operators are utilized for exploration mainly, which can be divided into ionization stage and fusion stage. To verify the feasibility and superiority of NRO, six unconstrained benchmark functions and a constrained engineering design problem are evaluated. These experiments demonstrate that the proposed NRO algorithm can effectively show the feasibility of solving the optimization problems and outperform other state-of-the-art algorithms.

ACKNOWLEDGMENTS

The study is supported by Aeronautical Science Foundation of China (No.20175196019) and Shaanxi nature science foundation (No.2017JM6078).

REFERENCES

[1] Xiangtao, L., Jie, Z., Minghao, Y. 2014. Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput & Applic 24:1867–1877 (2014). DOI=10.1007/s00521-013-1433-8
[2] S.J. Nanda, G. Panda. 2014. A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol. Comput. 16: 1 – 18 (2014).

[3] H. Zang, S. Zhang, K. Hapeshi. 2010. A review of nature-inspired algorithms. J. Bionic Eng. 7: 232 – 237 (2010).

[4] K. V Price, R. M. Storn, and J. A. Lampinen. 2005. Differential Evolution. Berlin/Heidelberg: Springer-Verlag.

[5] J. R. Sampson. 1976. Adaptation in Natural and Artificial Systems (John H. Holland). SIAM Rev. 18, 3: 529–530 (Jul. 1976).

[6] R. Eberhart, J. Kennedy. 1995. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE. 39-43(1995).

[7] M. Dorigo, M. Birattari, T. Stützle. 2006. Ant colony optimization. IEEE Comput. Intell. Mag. 1, 4: 28-39 (2006).

[8] B. Basturk, D. Karaboga. 2006. An artificial bee colony (ABC) algorithm for numeric function optimization. In proceedings of IEEE Swarm Intelligence Symposium. 8: 687-697(2006).

[9] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, 2010, pp. 65-74.

[10] Yang X-S, Deb S. 2010. Engineering optimisation by cuckoo search. Int J Math Model Numer Optim. 1:330-43 (2011).

[11] Mirjalili S, Mirjalili S M, Lewis A. 2014. Grey wolf optimizer. Adv Eng Softw. 69:46-61 (2014).

[12] S. Saremi, S. Mirjalili, A. Lewis. 2017. Grasshopper optimisation algorithm: theory and application. Adv. Eng. Software. 105: 30-47(2017).

[13] Seyedali Mirjalili, Amir H. Gandomi, Seyyed Zahra Mirjalili, Shahrzad Saremi, Hosssam Faris, Seyed Mohammad Mirjalili. 2017. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software. 1-29(2017).

[14] X. Qi, Y. Zhu, H. Zhang. A new meta-heuristic butterfly-inspired algorithm. Journal of Computational Science.

[15] S. Mirjalili, A. Lewis. 2016. The whale optimization algorithm, Adv. Eng. Software. 95: 51 – 67(2016).

[16] Mohit Jain, Vijander Singh, Asha Rani. 2018. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm and Evolutionary Computation. 1 – 28. DOI=https://doi.org/10.1016/j.swevo.2018.02.013

[17] Min-Yuan Cheng, Doddy Prayogo. 2014. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. Computers and Structures. 139, 98-112. DOI=http://dx.doi.org/10.1016/j.compstruc.2014.03.007

[18] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc. 2014. Spider Monkey Optimization algorithm for numerical optimization. Memetic Comput., 6(1): 31–47. (2014). DOI=10.1007/s12293-013-0128-0

[19] Rashedi E, Nezamabadi-pour H, Saryazdi S. 2009. GSA: a gravitational search algorithm. Intelligent Information Management. 4(6): 390 – 395(2009).

[20] Albert Y.S. Lam, Victor O.K. Li. 2010. Chemical-Reaction-Inspired Metaheuristic for Optimization. IEEE Transactions on evolutionary computation. 14(3) (2010).

[21] Hussain Shareef, Ahmad Asrul Ibrahim, Ammar Hussein Mutlag. 2015. Lightning search algorithm. Applied Soft Computing 36: 315–333(2015). DOI=http://dx.doi.org/10.1016/j.asoc.2015.07.028

[22] W. Zhao, L. Wang, and Z. Zhang, “A novel atom search optimization for dispersion coefficient estimation in groundwater,” Futur. Gener. Comput. Syst., 91: 601–610 (2019). DOI=https://doi.org/10.1016/j.future.2018.05.037

[23] F. Zou, L. Wang, X. Hei and D. Chen. 2015. Teaching-learning based optimization with learning experience of other learners and its application. Applied Soft Computing 37: 725-736 (2015).
[24] Mudong Li, Hui Zhao, Xingwei Weng, Tong Han. 2016. Cognitive behavior optimization algorithm for solving optimization problems. *Applied Soft Computing* 39: 199 – 222 (2016). DOI=http://dx.doi.org/10.1016/j.asoc.2015.11.015

[25] Seyed Jalaleddin Mousavirad, Hossein Ebrahimpour-Komleh. 2017. Human mental search: a new population-based metaheuristic optimization algorithm. *Appl Intell* 47: 850–887. DOI=10.1007/s10489-017-0903-6.

[26] Zhang Zhuoran, Huang Changqiang, Huang Hnaqiao, Tang Shangqin, Dong Kangsheng. 2018. An optimization method: hummingbirds optimization algorithm. *Journal of Systems Engineering and Electronics* 29(2): 386 – 404 (April 2018). DOI=10.21629/JSEE.2018.02.19