A brief overview of the synergy between metaheuristics and unconventional dynamics

Citation

ŠENKEŘÍK, Roman. A brief overview of the synergy between metaheuristics and unconventional dynamics. In: *Lecture Notes in Electrical Engineering* [online]. vol. 554, Springer Verlag, 2020, p. 344 - 356 [cit. 2023-02-02]. ISBN 978-3-03-014906-2. ISSN 1876-1100. Available at https://link.springer.com/chapter/10.1007/978-3-030-14907-9_34

DOI

https://doi.org/10.1007/978-3-030-14907-9_34

Permanent link

https://publikace.k.utb.cz/handle/10563/1008758

A Brief Overview of the Synergy Between Metaheuristics and Unconventional Dynamics

Roman Senkerik

Faculty of Applied Informatics, Tomas Bata University in Zlin, Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic senkerik@utb.cz

Abstract

This brief review paper focuses on the modern and original hybridization of the unconventional dynamics and the metaheuristic optimization algorithms. It discusses the concept of chaos-based optimization in general, i.e. the influence of chaotic sequences on the population diversity as well as at the metaheuristics performance. Further, the non-random processes used in evolutionary algorithms, and finally also the examples of the evolving complex network dynamics as the unconventional tool for the visualization and analysis of the population in popular optimization metaheuristics. This work should inspire the researchers for applying such methods and take advantage of possible performance improvements for the optimization tasks.

Keywords: Optimization, metaheuristics, evolutionary algorithms, complex networks, chaotic systems

1 Introduction

This brief survey explores the unconventional synergy of several different research fields belonging to the computational intelligence paradigm, which are the stochastics processes, complex chaotic dynamics, complex networks (CN), and metaheuristics algorithms, specifically evolutionary computation techniques (ECT's). The algorithms of the interest here are Differential Evolution (DE) [**1**], Particle Swarm Optimization (PSO) [**2**], Self Organizing Migrating Algorithm (SOMA) [**3**], Firefly Algorithm (FA) [**4**], and Firework algorithm (FWA) [**5**].

The motivation behind this survey is quite simple. In recent decades, the metaheuristic techniques became well-established and frequently used tools for solving engineering and research optimization tasks with a various level of complexity in both real and discrete domains. Despite the fact, that the ongoing research has brought many powerful and robust metaheuristic algorithms, the researchers have to deal with a well-known phenomenon so-called no free lunch theorem [**6**] forcing them to test various methods, techniques, adaptations and parameter settings leading to the acceptable results. The importance of finding a well-performing algorithm is growing together with the increase of the dimensionality and number of complex objectives in current optimization tasks. Moreover, it is necessary to emphasize the fact that, like most of these methods, they are inspired by natural evolution, and their development can be considered as a form of evolution. Such a fact is mentioned in the article [**7**] that even incremental steps in algorithm development, including failures, may be the inspiration for the development of robust and powerful metaheuristics. Also, it is always advisable to focus on simplifying algorithms, as stated in [**8**].

Thus this survey introduces several simple, yet successful, modifications and unconventional approaches applied for metaheuristic algorithms. This paper represents a follow-up and summarization of previous research [**9-12**]. The organization is the following: firstly, hybridization of chaotic dynamics and optimization algorithms is introduced, followed by the short chapter discussing the utilization of non-random processes. Finally, the original concept of the mutual connection between CN and evolutionary algorithms (EAs) is described.

2 Chaos Driven Metaheuristics

The key operation in metaheuristic algorithms is the randomness. Together with the persistent development of metaheuristic algorithms, the popularity of hybridizing them with deterministic chaos is growing every year, due to its properties like ergodicity, stochasticity, self-similarity, and density of periodic orbits. Also, vice-versa, the metaheuristic approach in chaos control/synchronization is more popular in recent years.

Recent research in chaotic approach for metaheuristics mostly uses straightforwardly various chaotic maps in the place of pseudo-random number generators (PRNG).

The original chaos-based approach is tightly connected with the importance of randomization within heuristics as compensation of a limited amount of search moves. This idea has been carried out in several papers describing different techniques to modify the randomization process [**13, 14**], as well as the influence of randomization operations to parameter adaptation was profoundly experimentally tested in [**15**].

The original concept of embedding chaotic dynamics into the evolutionary/swarm algorithms as chaotic pseudo-random number generator (CPRNG) is given in [**16**]. Firstly, the PSO algorithm with elements of chaos was introduced as CPSO [**17**], followed by the initial testing of chaos embedded DE [**18**], DE with chaotic mutation factor (SACDE) [19], and with the deterministic chaos for the initialization (CIDE algorithm) [**20**]. Original inertia weight based PSO strategy driven by CPRNGs was also profoundly investigated [**21-23**]. Besides the continuous space domain, chaos-driven metaheuristic proved to be successful also in the discrete domain [**24, 25**].

Recently the chaos driven heuristic concept has been utilized in several swarm-based algorithms [**26-30**], as well as many applications with DE [**31, 32**].

2.1 Chaos as the CPRNG

The general idea of CPRNG is to replace the default PRNG with the chaotic system (either discrete map or discretized time-continuous flow). Following nine well known and frequently studied discrete dissipative chaotic maps were used as the CPRNGs for various metaheuristics. Systems of the interest were: Arnold Cat Map, Burgers Map, Delayed Logistic Map, Dissipative Standard Map, Henon Map, Ikeda Map, Lozi Map, Sinai Map, and Tinkerbell Map. With the typical settings and definitions as in [**33**], systems exhibit typical chaotic behavior. Please refer to the Table **1** and Eqs. (1)-(9)

for the definition of the maps. Also, Fig. **1** shows the short chaotic sequences for all nine above mentioned maps. These plots support the claims that due to the presence of self-similar chaotic sequences, the heuristic is forced to neighborhood-based selection (or alternative communication in swarms).

Table 1. Definition of popular chaotic maps

Chaotic system	Notation	Parameters
Arnold Cat map	$X_{n+1} = X_n + Y_n (\text{mod } 1)$ $Y_{n+1} = X_n + kY_n (\text{mod } 1)$ (1)	$k = 2.0$
Burgers map	$X_{n+1} = aX_n - Y_n^2$ $Y_{n+1} = bY_n + X_n Y_n$ (2)	$a = 0.75$ and $b = 1.75$
Delayed Logistic	$X_{n+1} = AX_n(1 - Y_n)$ $Y_{n+1} = X_n$ (3)	$A = 2.27$
Dissipative Standard map	$X_{n+1} = X_n + Y_{n+1} (\text{mod } 2\pi)$ $Y_{n+1} = bY_n + k \sin X_n (\text{mod } 2\pi)$ (4)	$b = 0.1$ and $k = 8.8$
Hénon map	$x_{n+1} = a - x_n^2 + by_n$ $y_{n+1} = x_n$ (5)	$a = 1.4$ and $b = 0.3$
Ikeda map	$X_{n+1} = \gamma + \mu(X_n \cos\phi + Y_n \sin\phi)$ $Y_{n+1} = \mu(X_n \sin\phi + Y_n \cos\phi)$ (6) $\phi = \beta - \alpha/\left(1 + X_n^2 + Y_n^2\right)$	$\alpha = 6$, $\beta = 0.4$, $\gamma = 1$ and $\mu = 0.9$
Lozi Map	$X_{n+1} = 1 - a\lvert X_n\rvert + bY_n$ $Y_{n+1} = X_n$ (7)	$a = 1.7$ and $b = 0.5$
Sinai map	$X_{n+1} = X_n + Y_n + \delta \cos 2\pi Y_n (\text{mod } 1)$ $Y_{n+1} = X_n + 2Y_n (\text{mod } 1)$ (8)	$\delta = 0.1$
Tinkerbell map	$X_{n+1} = X_n^2 - Y_n^2 + aX_n + bY_n$ $Y_{n+1} = 2X_n Y_n + cX_n + dY_n$ (9)	$a = 0.9$, $b = -0.6$, $c = 2$ and $d = 0.5$

Obtaining the CPRNG output is quite simple to implement. As the chaotic system is a set of equations with a static start position (See Table 1), we created random start positions of the chaotic systems, to have different start position for different experiments. Thus we are utilizing the typical feature of chaotic systems, which is extreme sensitivity to the initial conditions, popularly known as "butterfly effect," as the random seed. This random position is initialized with the default implementation PRNG. Once the start position of the chaotic system has been obtained, the system generates the next sequence using its current position. Getting the normalized pseudo-random value from the typical range of 0-1 can be done in several ways:

- The abs value of current output iteration of the chaotic map (x-axis), is divided by the maximum value from generated chaotic series. This simple scaling approach is causing so-called folding of the attractor around y-axis.
- The whole generated chaotic series is shifted to the positive real number region and then normalized to the range of 0-1.

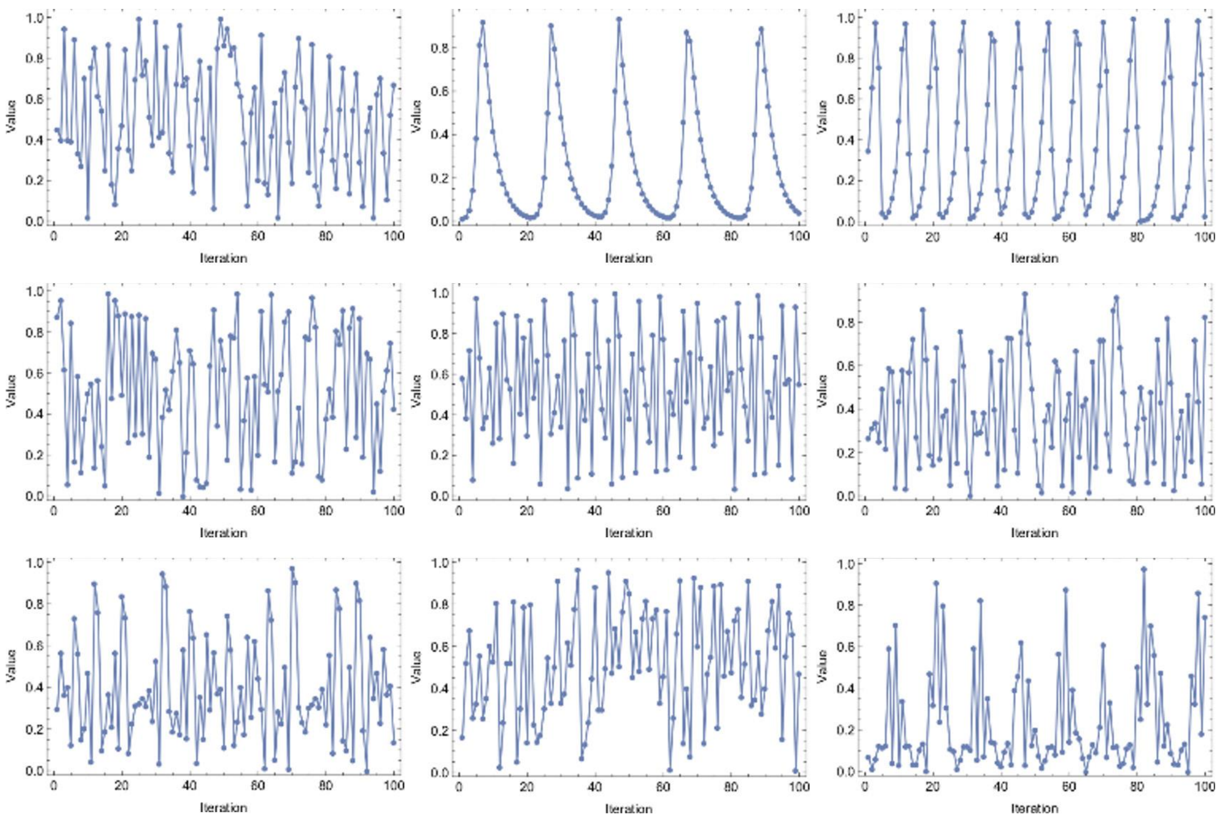

Fig. 1. Chaotic sequences normalized to the typical range of 0-1 for CPRNG; Arnold Cat map (upper left), Burgers Map (upper middle), and so on for the other maps (Delayed Logistic, Dissipative, Henon, Ikeda, Lozi, Sinai, Tinkerbell).

Also, the chaotic flows and oscillators have been widely studied, as well as other physical or chemical phenomenon showing chaos [**9**].

Nevertheless, most of the referred studies have used the chaotic dynamics to improve the properties of algorithms in a particular application, and unfortunately, deeper insights or theoretical explanations are not present in those research papers. The questions remain unanswered:

- Why does it work? Why may it be beneficial to use the chaotic sequences for pseudo-random numbers driving the selection, mutation, crossover or other processes in particular heuristics?
- Are there any chaotic features in the evolving population?

The first question was experimentally investigated in [**34**], and [**35**], where different sampling rates applied to the chaotic sequences were resulting in either keeping, or partially/fully removing of traces of chaos. These works show that not the distribution of the CPRNG used, but the unique sequencing given by the chaotic attractor hidden dynamics seems to be the key feature that may improve the heuristic performance.

In some instances, the population dynamics (which can also be considered as a chaotic or at the edge of chaos) seems to self-synchronize with the chaotic attractor dynamics and sequencing. This was not fully examined at all.

Based on the growing popularity of the complex/ensemble adaptation approaches within metaheuristics, the multi-chaotic CPRNGs were introduced in papers [**21**], and [**36**].

Moreover, as can be seen in Table **1**, chaotic systems have easy-accessible parameters, which can be tuned resulting in different CPRNG distributions, sequencing of chaotic series, thus different influence on the metaheuristics being driven, as studied in [37]. The findings from referred research papers can be summarized as follows:

- Above referred research papers confirmed that used optimization metaheuristic is sensitive to the chaotic dynamics driving the selection, mutation, or communication process inside swarms through CPRNG. At the same time, it is clear that (selection of) the best CPRNGs are problem-dependent. The observed performance of enhanced optimizer is (significantly) different: either better or worse against other compared versions. Such a worse performance was repeatedly observed for three chaotic maps: Delayed logistic, Burgers, and Tinkerbell. On the other hand, these maps usually secured very fast progress towards function extreme (local) followed by premature population stagnation phase, thus repeatedly secured the finding of minimum values [**12**].
- The multi-chaotic generators [**38**] or ensemble systems could be beneficial since the used metaheuristic algorithm can profit from the combined/selective population diversity (i.e., exploration/exploitation) tendencies, sequencing-based either stronger or moderate progress towards the function extreme, all given by the smart combination of multi-randomization schemes.
- Swarm algorithms seem to benefit more from chaotic dynamics than “classical” EAs like Genetic Algorithms (GA) and DE.
- The population diversity analysis presented in [**12**] supports the theory, that unique features of the chaos transformed into the sequencing of CPRNG values may create the subpopulations or inner neighborhood selection schemes. Thus the metaheuristic can benefit from the searching within those sub-populations and quasiperiodic exchanges of information between individuals.
- There are many unexplored aspects and theories, like an auto-parameter adaptation for attractors driving metaheuristics, synchronization with optimized (dynamical) systems, and many more.
- The complex sequencing given by the chaotic system used seems to be the essence, not just the different distribution of CPRNG.
- Although intensive benchmarking (CEC benchmark suites) showed mixed results, in real-life optimization problems, the chaos driven heuristics is performing very well [**39, 40**], especially for some instances in the discrete domain [**24, 25**] - with the exception mentioned below.
- Comparison of PRNGs and CPRNGs is given in [**41**], whereas a critical review is in [**42**].

2.2 Other Unconventional Approaches with Chaos

Besides the direct utilization of CPRNGs, there exist two more approaches for interconnection between chaos and metaheuristics.

- The complexity of chaotic systems and its movement in the space is used for dynamical mapping of the search space mostly within the local search techniques [**43**].
- Finally, the hybridization of searching/optimization process and chaotic systems is represented by chaos based random walk technique [**44**].

3 Non-random Processes and Evolutionary Algorithms

As stated in the introduction, an inherent part of EAs, are random processes. Interesting study [**10**] is discussing whether random processes are needed EAs. Simple experiments revealed the fact that random number generators can be replaced by deterministic processes with short periodicity. Authors are claiming, that an advantage of the deterministic processes utilization is the possibility to repeat the experiment, analysis of algorithm behavior, mapping of its full path on the searched fitness landscape, and finally the possibility of easier construction of any kind of (mathematical) proofs for the used class of the EAs.

Also, different examples of non-random processes can be found in [**45**], where the sinusoidal function is used within the control parameters adaptation process.

4 Metaheuristics and Complex Networks

In this chapter, that represents a follow-up of more detailed studies [**11, 46, 47**], a fusion of two different attractive areas of research: (complex) networks and evolutionary computation, is described. Interactions in a swarm/evolutionary algorithms during the optimization process can be considered like user interactions in social networks or just people in society. It has been observed that networks generated by evolutionary dynamics show properties of CN in certain time frames and conditions [**48**]. The CN approach is utilized to show the linkage between different individuals in the population. Each individual in the population can be taken as a node in the CN graph, where its links specify the successful exchange of information in the population.

The population is visualized as an evolving CN that exhibits non-trivial features -e.g., degree distribution, clustering, and centralities. These features are important markers for a population used in evolutionary/swarm-based algorithms and can be utilized for the adaptive population control as well as parameter control during the metaheuristic run. The initial studies [**49, 50**] describing the possibilities of transforming population dynamics into CN were followed by the successful adaptation and control of the metaheuristic algorithm during the run through the given CN frameworks [**51-55**].

This paper briefly reviews the CN frameworks for DE, PSO, FA, and FWA. All referred works have shared a common motivation:

- To show the different approaches in building CN to capture the dynamics either of evolutionary or swarm-based algorithms.
- To investigate the time development of the influence of either individual selection for the mutation/crossover process or communication inside a swarm transferred into the CN.
- To show the usability of CN attributes, that can be extracted from graph visualizations, for adaptive population and parameter control during the metaheuristic run.

Since the internal principles are different for “classical” evolutionary-based (DE, GA), and swarm-based algorithms (PSO, FA, and FWA), several different approaches for capturing the population dynamics have been developed and tested:

- Capturing of the evolution process, i.e., the contribution of individuals from the population (DE, GA).
- Capturing of the communication in the swarm algorithms (PSO, FA)

In the case of the classical EAs, mostly an Adjacency Graph was used. In each generation, the node is only active for the successful transfer of information, i.e., if the individual is successful in generating a new better individual who is accepted for the next generation of the population, one establishes the connections between the newly created individual and the (for the DE - several) sources; otherwise, no connections are recorded in the Adjacency Matrix. An illustrative example is given in Fig. **2** containing Adjacency Graph for the short snapshot (10 iterations at the beginning of the optimization process), and also the example of the corresponding community plot. The Degree Centrality value is highlighted by the size of the node (red color).

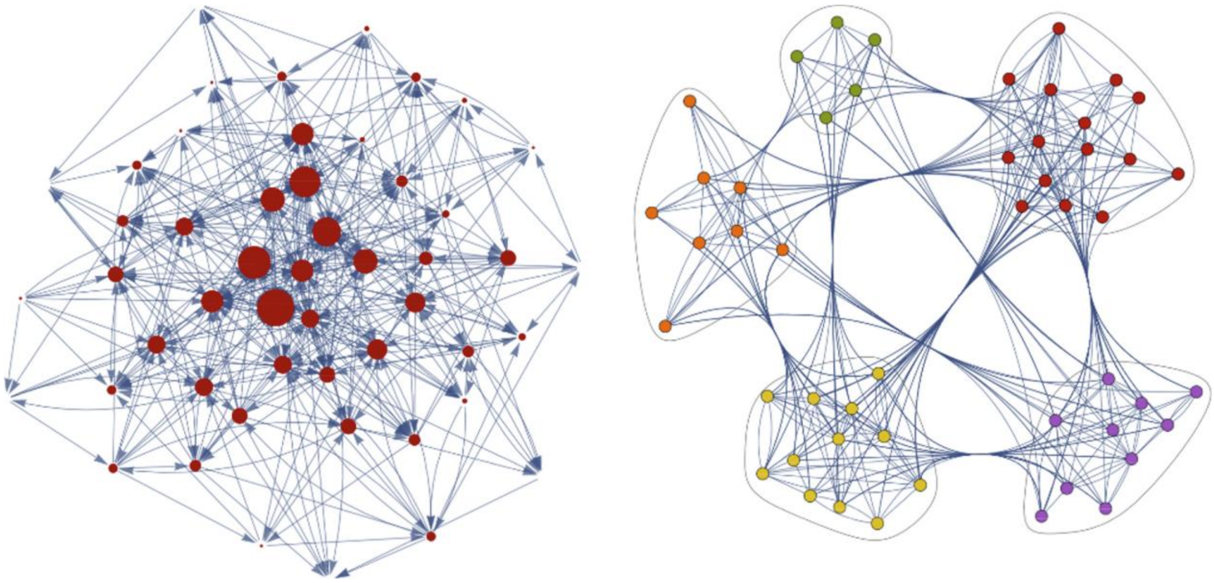

Fig. 2. Adjacency graph for a short time snapshot of DE algorithm (left), and corresponding community plot (right).

Analysis of CNs from DE algorithm can be found in [**51, 56-59**]; and also in a comprehensive study discussing the usability of network types [**60**].

The transition between evolutionary based and swarm-based algorithms can be seen in the FA. Although it is a swarm type, the situation here is very similar to the classical EAs. Every firefly is depicted as a node. The connection between nodes is plotted for every successful interaction when firefly flies towards another and improves own brightness. Details are discussed in [**61**].

For the PSO algorithm, the main interest is in the communications that lead to population quality improvement. Therefore, only communication leading to improvement of the particles personal best results (noted in original PSO as pBest) was tracked [**55**]. Another study was dealing with capturing of the density of communication [**62**], that can reveal the relations between the density of communication and convergence speed of the PSO.

Alternatively, it is possible to construct an Adjacency Graph and to benefit from its statistical features - as with the DE/GA/FA case. The direct link is created between the particle that has improved the global shared best solution and the particle that has been improved later based on such update. More investigation aimed at PSO and CN framework are in [**63**]. The very interesting approach of fitness landscape classifying based on the CN features and PSO is in [**64**].

The last studied algorithm (FWA) is the original representative of the random search/local search engine type algorithm. The paper [**65**] has shown, that even for this type, it is possible to develop a scheme for capturing the communication in the form of a graph. An interesting phenomenon has been discovered. The network seems to have a lack of any other usable information, besides the ability to identify the surface type of optimized function.

Besides the above-presented approaches, more have been explored for a wider portfolio of algorithms [**66-68**]. Overall, findings from the all referred papers can be summarized in the following way:

- The building of the Network: Since there is a direct link between parent solutions and offspring in the classical EAs, this information is used to build a CN. In the case of swarm algorithms, it depends on the inner swarm mechanisms, but mostly, it is possible to capture the communications within the swarm based on the points of attraction driven information updating.
- Any original approach leads to the different graph visualizations and possible subsequent analyses.
- Complex Network Features: Centralities and clustering/community analyses may contain direct information about the selection of individuals and their success; therefore, many network features can be used for controlling a population during an EA run. It is possible to use such information either for the injection or replacement of individuals or to modify/alternate the evolutionary strategy [**57, 58**]. In the case of swarm algorithms, the communication dynamics are captured - thus the level of particle performance (usefulness) can be calculated, or some sub-clusters and centralities of such communication can also be identified [**63**].
- Fitness Landscape: The capturing of communications (swarm dynamics) is sensitive to the fitness landscape. Thus, network features can be used for the raw estimation of a fitness landscape [**64**].
- Dimensional independence - since the size of the network is given only by the number of nodes (individuals from the populations), the resulting features analyses are not directly connected to the dimensionality of the search space.
- The advantage is that the CN framework can be used almost on any metaheuristic.

5 Conclusions

The primary aim of this original work is to provide more in-depth insights into the synergy between popular optimization metaheuristic algorithms and unconventional dynamics. The research of randomization issues and insights into the inner dynamic of metaheuristic algorithms was many times addressed as essential and beneficial. The influence of chaotic sequences on the metaheuristics performance, further the nonrandom processes used in EAs, and finally, the original concept of the evolving CN analysis for the better understanding of the population dynamics in popular optimization metaheuristics is briefly reviewed here. Important conclusions and findings are summarized at the end of each chapter.

References

1. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution-an updated survey. Swarm Evol. Comput. 27, 1-30 (2016)

2. Engelbrecht, A.P.: Heterogeneous particle swarm optimization. In: International Conference on Swarm Intelligence, pp. 191-202. Springer, Heidelberg, September 2010

3. Zelinka, I.: SOMA—self-organizing migrating algorithm. In: Self-Organizing Migrating Algorithm, pp. 3-49. Springer, Cham (2016)

4. Fister, I., Fister Jr., I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm Evol. Comput. 13, 34-46 (2013)

5. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: International Conference in Swarm Intelligence, pp. 355-364. Springer, Heidelberg, June 2010

6. Droste, S., Jansen, T., Wegener, I.: Perhaps not a free lunch but at least a free appetizer. In: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation, vol. 1, pp. 833-839. Morgan Kaufmann Publishers Inc., July 1999

7. Piotrowski, A.P., Napiorkowski, J.J.: Step-by-step improvement of JADE and SHADE-based algorithms: success or failure? Swarm Evol. Comput. 43, 88-108 (2018)

8. Piotrowski, A.P., Napiorkowski, J.J.: Some metaheuristics should be simplified. Inf. Sci. 427, 32-62 (2018)

9. Senkerik, R., Zelinka, I., Pluhacek, M.: Chaos-based optimization-a review. J. Adv. Eng. Comput. 1(1), 68-79 (2017)

10. Zelinka, I., Lampinen, J., Senkerik, R., Pluhacek, M.: Investigation on evolutionary algorithms powered by nonrandom processes. Soft. Comput. 22(6), 1791-1801 (2018)

11. Senkerik, R., Zelinka, I., Pluhacek, M., Viktorin, A.: Study on the development of complex network for evolutionary and swarm based algorithms. In: Mexican International Conference on Artificial Intelligence, pp. 151-161. Springer, Cham, October 2016

12. Senkerik, R., Viktorin, A., Pluhacek, M., Kadavy, T.: Population diversity analysis for the chaotic based selection of individuals in differential evolution. In: International Conference on Bioinspired Methods and Their Applications, pp. 283-294. Springer, Cham, May 2018

13. Weber, M., Neri, F., Tirronen, V.: A study on scale factor in distributed differential evolution. Inf. Sci. 181(12), 2488-2511 (2011)

14. Neri, F., Iacca, G., Mininno, E.: Disturbed exploitation compact differential evolution for limited memory optimization problems. Inf. Sci. 181(12), 2469-2487 (2011)

15. Zamuda, A., Brest, J.: Self-adaptive control parameters' randomization frequency and propagations in differential evolution. Swarm Evol. Comput. 25, 72-99 (2015)

16. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans. Evol. Comput. 7(3), 289-304 (2003)

17. Coelho, L.d.S, Mariani, V.C.: A novel chaotic particle swarm optimization approach using Hénon map and implicit filtering local search for economic load dispatch. Chaos, Solitons Fractals 39(2), 510-518 (2009)

18. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Comput. Math Appl. 60(4), 1088-1104 (2010)

19. Zhenyu, G., Bo, C., Min, Y., Binggang, C.: Self-adaptive chaos differential evolution. In: International Conference on Natural Computation, pp. 972-975. Springer, Heidelberg, September 2006

20. Ozer, A.B.: CIDE: chaotically initialized differential evolution. Expert Syst. Appl. 37(6), 4632-4641 (2010)

21. Pluhacek, M., Senkerik, R., Davendra, D.: Chaos particle swarm optimization with Eensemble of chaotic systems. Swarm Evol. Comput. 25, 29-35 (2015)

22. Pluhacek, M., Senkerik, R., Davendra, D., Oplatkova, Z.K., Zelinka, I.: On the behavior and performance of chaos driven PSO algorithm with inertia weight. Comput. Math Appl. 66(2), 122-134 (2013)

23. Pluhacek, M., Senkerik, R., Viktorin, A., Kadavy, T.: Chaos-enhanced multiple-choice strategy for particle swarm optimisation. Int. J. Parallel Emergent Distrib. Syst. 1-14 (2018)

24. Metlicka, M., Davendra, D.: Chaos driven discrete artificial bee algorithm for location and assignment optimisation problems. Swarm Evol. Comput. 25, 15-28 (2015)

25. Davendra, D., Bialic-Davendra, M., Senkerik, R.: Scheduling the lot-streaming flowshop scheduling problem with setup time with the chaos-induced enhanced differential evolution. In: 2013 IEEE Symposium on Differential Evolution (SDE), pp. 119-126. IEEE, April 2013

26. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Firefly algorithm with chaos. Commun. Nonlinear Sci. Numer. Simul. 18(1), 89-98 (2013)

27. Wang, G.G., Guo, L., Gandomi, A.H., Hao, G.S., Wang, H.: Chaotic krill herd algorithm. Inf. Sci. 274, 17-34 (2014)

28. Zhang, C., Cui, G., Peng, F.: A novel hybrid chaotic ant swarm algorithm for heat exchanger networks synthesis. Appl. Thermal Eng. 104, 707-719 (2016)

29. Jordehi, A.R.: Chaotic bat swarm optimisation (CBSO). Appl. Soft Comput. 26, 523-530 (2015)

30. Wang, G.G., Deb, S., Gandomi, A.H., Zhang, Z., Alavi, A.H.: Chaotic cuckoo search. Soft. Comput. 20(9), 3349-3362 (2016)