Phase Transitions in Swarm Optimization Algorithms

Tomáš Vantuch[1(✉)], Ivan Zelinka[1], Andrew Adamatzky[2], and Norbert Marwan[3]

[1] Department of Computer science, Technical University of Ostrava, Ostrava, Czech Republic
tomas.vantuch@vsb.cz
[2] Unconventional Computing Lab, UWE, Bristol, UK
[3] Transdisciplinary Concepts and Methods, Potsdam Institute for Climate Impact Research (PIK), Potsdam, Germany

Abstract. Natural systems often exhibit chaotic behavior in their space-time evolution. Systems transiting between chaos and order manifest a potential to compute, as shown with cellular automata and artificial neural networks. We demonstrate that swarms optimisation algorithms also exhibit transitions from chaos, analogous to motion of gas molecules, when particles explore solution space disorderly, to order, when particles follow a leader, similar to molecules propagating along diffusion gradients in liquid solutions of reagents. We analyse these 'phase-like' transitions in swarm optimization algorithms using recurrence quantification analysis and Lempel-Ziv complexity estimation. We demonstrate that converging and non-converging iterations of the optimization algorithms are statistically different in a view of applied chaos, complexity and predictability estimating indicators.

Keywords: Chaos · Recurrence · Complexity · Swarm · Convergence

1 Introduction

Natural systems not rarely undergo phase transition when performing a computation (as interpreted by humans), e.g. reaction-diffusion chemical systems produce a solid precipitate representing geometrical structures [10], slime mould transits from a disorderly network of 'random scouting' to a prolonger filaments of protoplasmic tube connecting source of nutrients [2], 'hot ice' computer crystallizes [1]. Computation at the phase transition between chaos and order was firstly studied by Crutchfield and Young [12], who proposed measures of complexity characterising the transition. The ideas were applied to cellular automata by Langton [19]: a computation at the edge of chaos occurs due to gliders. Phase transitions were also demonstrated for a genetic algorithm which fall into a chaotic regime for some initial conditions [24,31] and network traffic models [25].

S. Stepney and S. Verlan (Eds.): UCNC 2018, LNCS 10867, pp. 204–216, 2018.
https://doi.org/10.1007/978-3-319-92435-9_15

Algorithmic models of evolutionary based optimization, AI and ALife possess comparable features of the systems with a higher complexity, they simulate [14, 36]. We focus on the behavioral modes: the presence of a random or pseudo-random cycling (analogous to gaseous phase state), ordered or a stable states (analogous to solid state), or the chaotic oscillations (transitive states). Each of the modes could imply different level of a computational complexity or an algorithm performance as it was revealed on different algorithms [6,7,15]. By detecting such modes we can control and dynamically tune performance of the computational systems.

A swarm-like behavior has been extensively examined in studies of Zelinka et al. [35] where the changing dynamics of an observed algorithm was modeled by a network structure. The relevance between network features and algorithm behavior supported the control mechanism that was able to increase the algorithm performance [30]. An extensive empirical review of existing swarm based algorithms has been brought by Schut [28] where approaches like collective intelligence, self-organization, complex adaptive systems, multi-agent systems, swarm intelligence were empirically examined and confronted with their real models which reflected several criteria for development and verification.

We aim to evaluate the dynamics of optimization algorithms, inspired by evolution and swarm-like behavior. We evaluate the dynamical modes of algorithms based on predictability, complexity and chaos features. At the end, we statistically examine the difference between estimated modes, they possessed. In case of successful detection of statistically different modes and their transitions during the optimization process, the edge of chaos may be examined as well as controlling tools may be designed. Having these tools may increase the ability to control the optimization process being on maximal convergence level.

2 Theoretical Background

2.1 Swarm Based Optimization

The optimization algorithms examined in our study are representatives of bio-inspired single-objective optimization algorithms. They iteratively maintain the population of candidates migrating through the searched space. Their current position represents the solution vector X of the optimized problem.

Particle Swarm Optimization implies that the combined particle's aim towards the global leader and its previous best position [17]. The composition of these two stochastically altered directions modifies its current position in order to find a better optimum of the given function. Several reviewing studies are available as extensive descriptions of the algorithm and they are also surveying proposed extensions and variations [4,13].

Differential Evolution (DE) was developed by Storn and Price [29] and it possesses the features of a self-organizing search as well as an evolutionary based optimization. This interconnection is deserved due to its three main stages. DE offers several strategies driving the computation of new positions for its candidates. One of them takes three random candidates to calculate an intermediate

candidate which creates a new position by binary crossover with an optimized candidate x_i. It takes this new position only if it is better than the current one.

Self-organizing migrating algorithm (SOMA) is a stochastic evolutionary algorithm was proposed by Zelinka [34]. Ideologically, this algorithms stands right between purely swarm optimization driven PSO and evolutionary-like DE. The entire nature of migrating individuals across the search-space is represented by steps in the defined path length and a stochastic nature of a perturbation parameter that represents specific version of the mutation. The perturbation creates binary vector by the adjusted PRT parameter and the given formula

$$v_j^{prt} = \begin{cases} 1, & \text{if } r_j < PRT \\ 0, & \text{otherwise} \end{cases}, (j = 1, 2, \cdots, d) \tag{1}$$

Applying V^{prt}, the path is perturbed towards new solution using current particle's and leaders position.

$$x_i^{t+1} = x_i^t + (x_L^t - x_i^t)v_i^{prt} \tag{2}$$

During each migration loop, each particle performs n steps according to the adjusted step size and the path length. If the path length is higher than one, particle will travel longer distance, than is his distance towards the leader.

2.2 Lemplel-Ziv Complexity

According to the Kolmogorov's definition of complexity, the complexity of an examined sequence X is the size of a smallest binary program that produces such sequence [11]. Because this definition is way too general and any direct computation is not guaranteed within the finite time [11], approximative techniques are frequently employed.

Lempel and Ziv designed a complexity estimation in a sense of Kolmogorov's definition, but limiting the estimated program only to two operations: recursive copy and paste [21]. The entire sequence based on an alphabet $\aleph$ is split into a set of unique words of unequal lengths, which is called a vocabulary. The approximated binary program making use of copy and paste operations on the vocabulary, is able to reconstruct the entire sequence. Based on the size of vocabulary ($c(X)$), the complexity is estimated as $C_{LZ}(X) = c(X)(\log_k c(X) + 1) \cdot N^{-1}$, where k means the size of the alphabet and N is the length of the input sequence. A natural extension for multi-dimensional LZ complexity was proposed in [37]. In case of a set of l symbolic sequences $X^i (i = 1, \cdots, l)$, Lempel and Ziv's definitions remain valid if one extends the alphabet from scalar values x_k to l-tuples elements $(x_k^1, \cdots, x_k^l)$. The joined-LZC is than calculated as $C_{LZ}(X^1, \cdots, X^l) = c(X^1, \cdots, X^l)(\log_{k^2} c(X^1, \cdots, X^l) + 1) \cdot N^{-1}$.

2.3 Recurrence Quantification Analysis

The recurrence plot (RP) is the visualization of the recurrence matrix of m-dimensional system states $\boldsymbol{x} \in \mathbb{R}^m$ [23]. The closeness of these states for

a given trajectory $\boldsymbol{x}_i$ $(i = 1, 2, ..., N)$ where N is the trajectory length, is thresholded in the Heaviside step function $\Theta(\cdot)$ which results in the binary matrix of recurrence $R_{i,j}(\epsilon) = \Theta(\epsilon - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$. The Euclidean norm is the most frequently applied distance metric $\|\cdot\|$ and the threshold value ϵ can be chosen according to several techniques [18,23,27,32,33].

If only one-dimensional time series is given, the phase space trajectory has to be reconstructed from the time series $\{u_i\}_{i=1}^{N}$, e.g., by using the time-delay embedding $\boldsymbol{x}_i = (u_i, u_{i+\tau}, ..., u_{i+(m-1)\tau})$, where m is the embedding dimension and τ is the embedding delay [26]. The parameters m and τ may be found using methods based on false nearest neighbors and auto-correlation [16].

The RQA measures applied in this experiment describe the predictability and level of chaos in the observed system. Determinism is defined as the percentage of points that form diagonal lines (Eq. 3)

$$DET = \sum_{l=2}^{N} lP(l) \left[\sum_{l=1}^{N} lP(l) \right]^{-1} \tag{3}$$

where $P(l)$ is the histogram of the lengths l of the diagonal lines [23]. Its values, ranging between zero and one, estimate the predictability of the system.

Divergence is related to the sum of the positive Lyapunov exponents, naturally computing the amount of chaos in the system, and it is defined as follows

$$DIV = L_{\max}^{-1}, \qquad L_{\max} = \max(\{l_i; i = 1, \cdots, N_l\}) \tag{4}$$

where $L_{\max}$ is the longest diagonal line in the RP (excluding the main diagonal line) [23].

3 Experiment Design

Data Preparation. All three examined algorithms attempted to optimize one common fitness-function, the Rastrigin function, because of its frequent application with similar manners and its dimensional scalability that satisfies our testing purposes: $f(x) = A \cdot n + \sum_{i=1}^{n} (x_i^2 - A \cdot \cos(2\pi x_i))$, where $A = 10$ and $x_i \in [-5.12, 5.12]$. The function has a global minimum at $x = 0$ where $f(x) = 0$.

The adjustment of the optimization algorithms was tuned by random search hyper-parameter optimization [5] in order to find the optimal adjustment to perform the best possible convergence. The only fixed hyper-parameters were the dimension of the optimized function (it also affected the dimension of the particles, $D = 10$) and the population size of the algorithm ($NP = 40$, 60, 100- it varied in order to see the affect of population size on the appearing dynamics). The rest of the hyper-parameters were optimized in the ranges according to Table 1.

The behavior of the optimization algorithms is represented by the positions $(X_{t_1} = \{x_{t_1,1}, x_{t_1,2}, \cdots, x_{t_1,D}\})$ taken by their population members $(P = p_1, p_2, \cdots, p_N)$ during their migrations/iterations $(p_1 = X_{t_1,1}, X_{t_2,1}, \cdots, X_{t_m,1})$. All of them are stored for the further examination. The time windows w of

iterations are taken and transfered into matrices of particles positions where columns are particle's coordinates and rows are ordered particles by their population number and time.

$$(P_{w_i} = \{x_{t_i,1}, x_{t_i,2}, \cdots, x_{t_i,N}, x_{t_{i+1},1}, x_{t_{i+1},2} \cdots, x_{t_{i+1},N}, \cdots x_{t_{i+w},N}\}).$$

Table 1. The value ranges of hyper-parameters of optimization algorithms to be adjusted with their meaning.

Parameter	Algorithm	Meaning	Value
c_1	PSO	global best position multiplier	$\langle 0.5, 1.5 \rangle$
c_2	PSO	local best position multiplier	$\langle 0.5, 1.5 \rangle$
w	PSO	inertia weight	$\langle 0.5, 0.95 \rangle$
F	DE	differential weight	$\langle 0.1, 1.0 \rangle$
Cr	DE	crossover probability	$\langle 0.1, 1.0 \rangle$
prt	SOMA	pertubation probability	$\langle 0.1, 1.0 \rangle$
step size	SOMA	size of the performed step	$\langle 0.1, 1.0 \rangle$

Convergence. Applying the before-mentioned algorithms' hyper-parameters, the optimization converged towards an optimum. In case of our experiment, the exclusive finding of a global optimum does not play such an important role as the fact that algorithms converge towards a fixed point performing various changes and interactions inside of their swarm. Various visualization settings (window size, population size) were tested in order to plot the most kinds of phase shifting behaviors. Figures, depicted as follows (Figs. 2, 3, 4), performed visually the representatives of the most common kinds.

The changes and interactions inside of their migrating populations are not usually visible in convergence plots, however changes during the convergence may be estimated using recurrence plots. For this purpose, three selected windows of algorithms' iterations were visualized to spot the differences among them. Figure 1 illustrates how phases of the algorithm convergences are reflected in recurrence plots.

Complexity Estimation. The obtained matrix P_{w_i} served as input for a joint Lempel-Ziv complexity (LZC) estimation and RQA. For the purpose of joint LZC estimation, the input matrix was discretized into adjustable number of letters n_l of an alphabet by the given formula. Let $p_{\min} = \min\{p_j | 1 \leq j \leq w\}$, $p_{\max} = \max\{p_j | 1 \leq j \leq w\}$ and $p_d = p_{max} - p_{min}$ then each element p_j is assigned value $p_j \leftarrow \lfloor n_l \frac{p_j - p_{min}}{p_d} \rfloor$. The joint-LZC therefore stands, in our case, for the complexity of time ordered n dimensional tuples (populations).

In case of RQA, there is a possibility to directly use the spatial data representation [22], therefore we did not apply the Takens' embedding theorem and we directly calculated the thresholded similarity matrix from our source data. The RQA features like determinism and divergence were calculated.

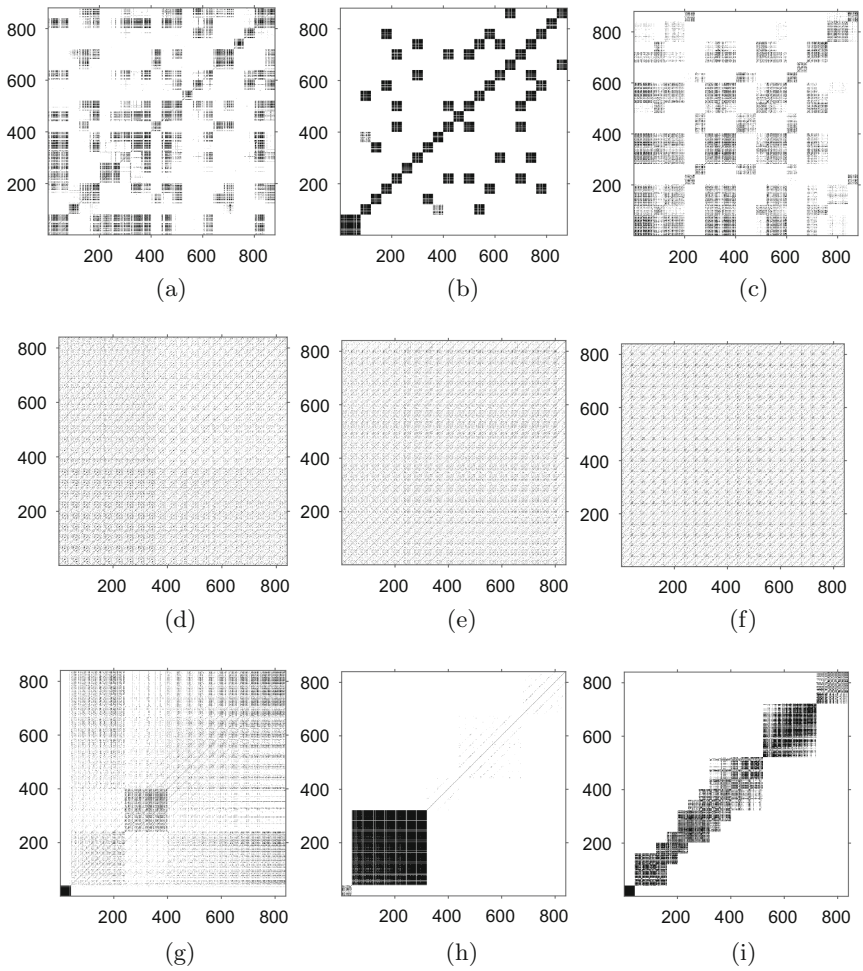

Fig. 1. Recurrence plots of the PSO (abc), DE (def), and SOMA (ghi) behavior calculated as similarities among the particles' positions X_t grouped into the windows of populations P_{w_i} during their (adg) "post-initial" (10th migration), (beh) "top-converging" (60th migration) and (cfi) "post-converging" (400th migration) phase.

Based on the obtained visualizations (Figs. 2, 3 and 4) we are able to confirm the visible differences in cases of PSO and SOMA algorithm. These two optimizations are performing similarities when the population is migrating the same direction. Once the optimum is reached, the similarities decrease. We are not able to confirm the same in case of DE. Due to the randomly performed crossover and additional mutation, this algorithm seems to contain more randomness and evolution-like behavior.

Further examinations calculated the DET, DIV and LZC values during all of the migrations. The statistical difference of these complexity indicators among the converging and non-converging iterations will be examined by ANOVA to confirm the presence of state transitions [20].

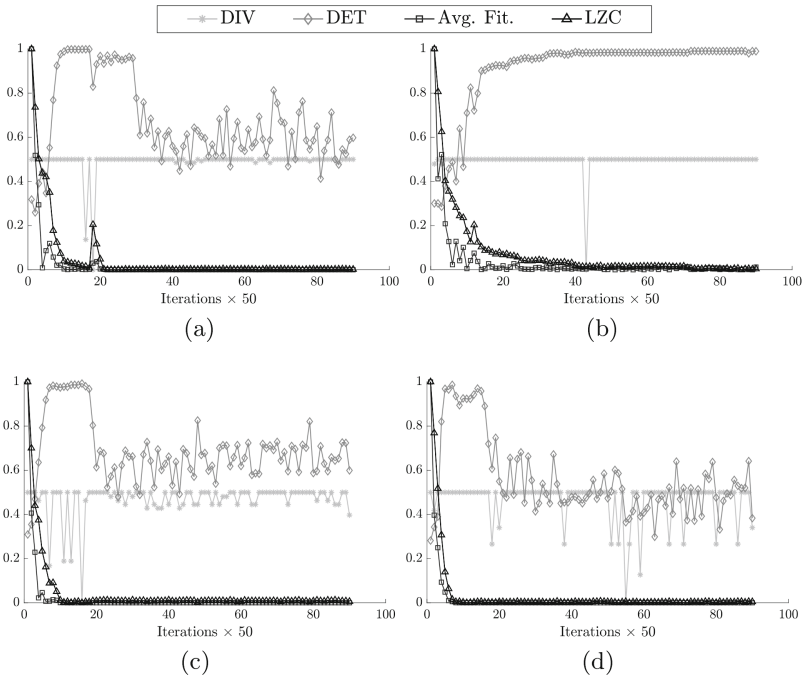

Fig. 2. Progress of the PSO algorithms executed several times with varying populations and window sizes. Horizontal axis represents the migrations while the vertical line holds values of average fitness-function of the population (Avg. Fit.) and obtained indicators. (a) population size 40, window 20, (b) population size 70, window 20, (c) population size 100, window 30, (d) population size 100, window 40

4 Results

Levels of complexity and the RQA indicators may posses different values based on a given window size as well as the size of the population, therefore we tried several combinations of these parameters (3 per each, therefore nine combinations for each algorithm). Only each tenth value of each time set was plotted in the charts (see Figs. 2, 3 and 4). The values of fitness-function and LZ complexity were normalized into the range between 0 and 1. The determinism returns such normalized values originally, therefore there was no need for an additional normalization. In case of the divergence, its values were very low ($\times$E10^{-3}), so it was necessary to multiply them in order to keep the similar visual scale in charts.

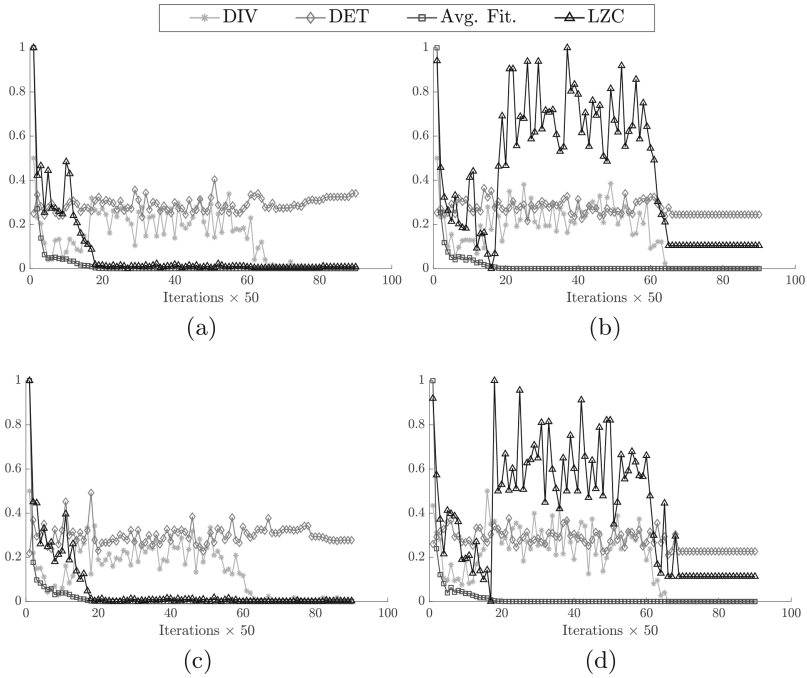

Fig. 3. Progress of the DE algorithms executed several times with varying populations and window sizes. Horizontal axis represents the migrations while the vertical line holds values of average fitness-function of the population (Avg. Fit.) and obtained indicators. (a) population size 40, window 20, (b) population size 70, window 20, (c) population size 70, window 30, (d) population size 70, window 40

Particle Swarm Optimization. The progress of PSO (Fig. 2) possess quickly decreasing LZC as the population converges towards an optimum and looses diversity. This behavior is expected as well as some appearing pulses in times when population probably left a local optimum, which was also reflected by an additional converges towards some better solution.

The progress of the population was very much predictable as it was evaluated by DET which possesed values close to 1 when the convergence of the population was the highest. Once a found optimum was reached by the majority of the population, DET dropped and evaluated the population's progress as unpredictable.

Higher values of DIV imply the presence of chaotic behavior in the system. All of the evaluations returned only very small values of this indicator therefore the only small amount of chaos can be confirmed. In the available visual evaluation, the DIV appears to possess the smallest relation to the progress of the algorithm.

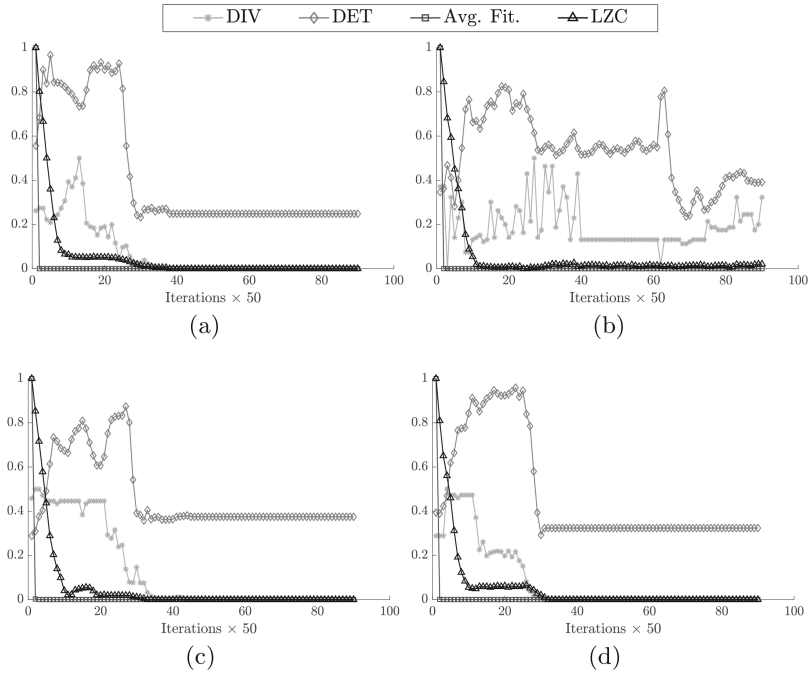

Fig. 4. Progress of the SOMA algorithms executed several times with varying populations and window sizes. Horizontal axis represents the migrations while the vertical line holds values of average fitness-function of the population (Avg. Fit.) and obtained indicators. (a) population size 40, window 20, (b) population size 40, window 30, (c) population size 40, window 40, (d) population size 100, window 20

Differential Evolution. DE performs elitism during its operation which can be the reason of an absolute flat progress of all its indicators during last iterations. The significant increase of LZC values in some cases remains unclear and can be connected with situation when the population found several optimums of the same quality and the population randomly switched among them (see Fig. 3). The values of DET only evaluate the entire progress of DE as unpredictable almost the same way as the DIV which marked the behavior as chaotic until the found optimum was reached by the population and any other better solution was found.

Self-Organizing Migration Algorithm. The progress of the SOMA algorithm has similarities with both previous algorithms. All indicators are very flat during its last migrations, because particles remains on their positions in cases when better solution was not found. The pertubet following of the leader is similarly reflected by DET as it was in case of PSO, when the behavior of the algorithm was marked as predictable until the majority of the population reached the found optimum. The appearance of the chaos is very low the same way as it was in previous

cases (DIV). The LZC as well as the Fitness dropped very quickly because of the nature of SOMA. Each particle performed multiple trials (steps as the path length divided by the step size) and the each population's individual migrated towards its best trial. This is the nature of the algorithm and the reason why it appears as the algorithm with the highest performance in the frame of our experiments.

4.1 ANOVA Testing

The DET, DIV and LZC values were split into values obtained in different phases of the optimization. Six groups, marked from 1 to 6, were defined by visual estimation as follows.

- **1** as progress of PSO algorithm during its converging migrations [10, 60]
- **2** as progress of DE algorithm during its converging migrations [10, 60]
- **3** as progress of SOMA algorithm during its converging migrations [10, 60]
- **4** as progress of PSO algorithm during its non-converging migrations [300, 350]
- **5** as progress of DE algorithm during its non-converging migrations [300, 350]
- **6** as progress of SOMA algorithm during its non-converging migrations [300, 350]

The presence of statistically significant differences among the means of these groups will confirm the state transitions. Especially we are interested whether the groups of the same algorithms are different and in which indicators.

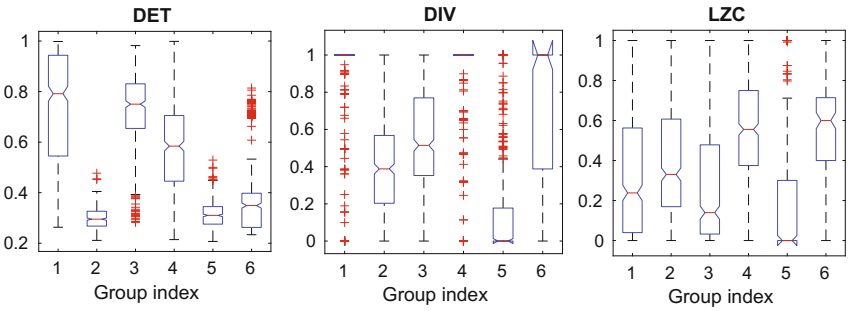

Fig. 5. Means with standard deviations obtained by ANOVA testing on six defined groups of data.

ANOVA testing rejected the null hypothesis that says about similarity of the means across the examined groups of the data (see Fig. 5). Obtained p-values are 0 for ANOVA_{DET} and ANOVA_{DIV}, and $2.657e-94$ for ANOVA_{LZC}. The performed additional post-hoc analysis revealed the specific differences among the groups according to their means and it is as follows. The means of Determinism

results were able to differentiate the groups 1 and 3 from the rest of the groups, while the means of second group were not significantly different from others (5, 6). The separability performance of the means of Divergence were able to significantly exclude the groups 2 and three from the rest while the means of the first group were similar to the fourth group. Both of them differed from the rest significantly. In case of LZC, the groups 1 and 3 are have means significantly different from the rest of the groups while group 2 possesses this difference against all of the groups.

These results mean that optimization phases are distinguishable by means of this complexity measure. From the above mentioned differences of the means, it is clearly visible that the convergence phases of PSO are separable by the means of Determinism and LZC while in case of Divergence we are not able to distinguish among them. In case of DE, its LZC and Divergence means possessed significant differences between DEs' convergence phases while Determinism was not applicable for this task. And finally the case of SOMA. All of the applied complexity criteria returned significantly different means among the SOMA convergence phases, therefore they are able to be distinguished by these values.

5 Discussion

In contrast to conventional computers, natural systems never stop to function, therefore by simply observing a physical, chemical or living computer we might never know when its completed the task and produced result. This phenomenon was formalized in a framework of inductive Turing machines [8] and advanced in structural machines [9], however still there is a lack of a definite measure. Some measures of spatio-temporal dynamics of a computing system are necessary to infer weather consider its current state as representing a final solution or wait longer.

In computer experiments with particle swarm optimization we found that it is possible to detect the convergence of algorithm using RQA and LZ complexity measures. The converging and non-converging iterations of the optimization algorithms are statistically different in the view of applied chaos, complexity and predictability estimating indicators. Typically, the degree of RQA Determinism sharply increases, as if undergoing a phase transition, when fitness approaches its maximum. Dynamics of LZ complexity follows, in general, the level of fitness. These results are well in line, and somewhat complement, our previous studies on the use of dynamics of compressibility of a system's spatial configurations to detect when the system completed computation [3].

Our findings may lead to the future work which is related to the estimation of the edge of chaos in the swarm-like optimization algorithms. It may be applied in a design of adaptive approaches aiming to control their progress in order to sustain the best possible performance.

Acknowledgment. The following grants are acknowledged for the financial support provided for this research: Grant of SGS No. SGS 2018/177, VSB-Technical University of Ostrava and German Research Foundation (DFG projects no. MA 4759/9-1 and MA4759/8).

References

1. Adamatzky, A.: Hot ice computer. Phys. Lett. A **374**(2), 264–271 (2009)
2. Adamatzky, A.: Advances in Physarum Machines: Sensing and Computing with Slime Mould., vol. 21. Springer, Switzerland (2016). https://doi.org/10.1007/978-3-319-26662-6
3. Adamatzky, A., Jones, J.: On using compressibility to detect when slime mould completed computation. Complexity **21**(5), 162–175 (2016)
4. Banks, A., Vincent, J., Anyakoha, C.: A review of particle swarm optimization. part i: background and development. Nat. Comput. **6**(4), 467–484 (2007)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
6. Bertschinger, N., Natschläger, T.: Real-time computation at the edge of chaos in recurrent neural networks. Neural Comput. **16**(7), 1413–1436 (2004)
7. Boedecker, J., Obst, O., Lizier, J.T., Mayer, N.M., Asada, M.: Information processing in echo state networks at the edge of chaos. Theory Biosci. **131**(3), 205–213 (2012)
8. Burgin, M.: Inductive turing machines with a multiple head and kolmogorov algorithms. Sov. Math. Dokl. **29**, 189–193 (1984)
9. Burgin, M., Adamatzky, A.: Structural machines and slime mould computation. Int. J. Gen. Syst. **42**, 1–24 (2017)
10. Costello, B.D.L., Adamatzky, A.: Calculating voronoi diagrams using chemical reactions. In: Advances in Unconventional Computing, pp. 167–198. Springer, Switzerland (2017). https://doi.org/10.1007/978-3-319-33921-4
11. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Hoboken (2012)
12. Crutchfield, J.P., Young, K.: Computation at the onset of chaos. In: The Santa Fe Institute, Westview, Citeseer (1988)
13. Del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans. Evol. Comput. **12**(2), 171–195 (2008)
14. Detrain, C., Deneubourg, J.L.: Self-organized structures in a superorganism: do ants "behave" like molecules? Phys. Life Rev. **3**(3), 162–187 (2006)
15. Kadmon, J., Sompolinsky, H.: Transition to chaos in random neuronal networks. Phys. Rev. X **5**(4), 041030 (2015)
16. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. University Press, Cambridge (1997)
17. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
18. Koebbe, M., Mayer-Kress, G.: Use of recurrence plots in the analysis of time-series data. In: SFI Studies in the Sciences of Complexity, vol. 12, pp. 361–361. Addison-Wesley Publishing (1992)
19. Langton, C.G.: Computation at the edge of chaos: phase transitions and emergent computation. Physica D **42**(1–3), 12–37 (1990)
20. Larson, M.G.: Analysis of variance. Circulation **117**(1), 115–121 (2008)
21. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Trans. Inf. Theory **22**(1), 75–81 (1976)
22. Marwan, N., Kurths, J., Saparin, P.: Generalised recurrence plot analysis for spatial data. Phys. Lett. A **360**(4), 545–551 (2007)

23. Marwan, N., Romano, M.C., Thiel, M., Kurths, J.: Recurrence plots for the analysis of complex systems. Phys. Rep. **438**(5), 237–329 (2007)
24. Mitchell, M., Hraber, P., Crutchfield, J.P.: Revisiting the edge of chaos: evolving cellular automata to perform computations. arXiv preprint adap-org/9303003 (1993)
25. Ohira, T., Sawatari, R.: Phase transition in a computer network traffic model. Phys. Rev. E **58**(1), 193 (1998)
26. Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw, R.S.: Geometry from a time series. Phys. Rev. Lett. **45**(9), 712 (1980)
27. Schinkel, S., Dimigen, O., Marwan, N.: Selection of recurrence threshold for signal detection. Eur. Phys. J. Spec. Top. **164**(1), 45–53 (2008)
28. Schut, M.C.: On model design for simulation of collective intelligence. Inf. Sci. **180**(1), 132–155 (2010)
29. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
30. Tomaszek, L., Zelinka, I.: On performance improvement of the soma swarm based algorithm and its complex network duality. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4494–4500. IEEE (2016)
31. Wright, A.H., Agapie, A.: Cyclic and chaotic behavior in genetic algorithms. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, pp. 718–724. Morgan Kaufmann Publishers Inc. (2001)
32. Zbilut, J.P., Webber, C.L.: Embeddings and delays as derived from quantification of recurrence plots. Phys. Lett. A **171**(3–4), 199–203 (1992)
33. Zbilut, J.P., Zaldivar-Comenges, J.M., Strozzi, F.: Recurrence quantification based liapunov exponents for monitoring divergence in experimental data. Phys. Lett. A **297**(3), 173–181 (2002)
34. Zelinka, I.: Soma–self-organizing migrating algorithm. In: New optimization Techniques in Engineering, vol. 141, pp. 167–217. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39930-8_7
35. Zelinka, I., Tomaszek, L., Vasant, P., Dao, T.T., Hoang, D.V.: A novel approach on evolutionary dynamics analysis-a progress report. J. Comput. Sci. **37**(5), 739–749 (2017)
36. Zenil, H., Gauvrit, N.: Algorithmic cognition and the computational nature of the mind. In: Encyclopedia of Complexity and Systems Science, pp. 1–9 (2017)
37. Zozor, S., Ravier, P., Buttelli, O.: On lempel-ziv complexity for multidimensional data analysis. Phys. A **345**(1), 285–302 (2005)