EVOLUTIONARY DYNAMICS CONTROL VIA ITS CML CONVERSION

Lukas Tomaszek
Department of Computer Science, FEI
VSB Technical University of Ostrava
Tr. 17. Listopadu 15, Ostrava
Czech Republic
lukas.tomaszek@vsb.cz

Ivan Zelinka
Department of Computer Science, FEI
VSB Technical University of Ostrava
Tr. 17. Listopadu 15, Ostrava
Czech Republic
ivan.zelinka@vsb.cz

Abstract
In this paper, is discussed control of CML system, that is derived from dynamics of evolutionary process-algorithm, i.e. CML used in our experiments reflect dynamics of evolutionary algorithm. The reflection is done so the dynamics of evolution is converted to the network and network to CML. Evolutionary dynamics, used here for control, is based differential evolution algorithm. The aim is to show, that dynamics of evolutionary processes can be modeled, visualized and controlled as the CML system is. Control process can be done by classical approach as well as by evolutionary algorithm. This paper discuss mainly visualization of evolutionary dynamics, its simple control and how does it change under pinning signal in order to prepare background for the new and full control experiments.

Key words
evolutionary algorithm, control, network, CML,

1 Introduction

Evolutionary algorithms (EA), copy natural processes, described by Ch. Darwin and G. Mendel. They are of discrete nature and can be used to solve computationally very hard problems. Since 70. they are an important part of modern engineering and computer science. The performance of EAs is one of the important issues of this algorithm technology. In our research (Zelinka, Tomaszek and Kojecky, 2016), (Senkerik, Viktorin, Pluhacek, Janostik and Oplatkova, 2016), (Zelinka, 2016) or (Zelinka, 2015) we proposed how can be EA converted into complex network (CN) and consequently to the coupled map lattices (CML) (Schöll and Schuster, 2008), (Zelinka, 2015), Fig. 1. In our research, we are attempting to convert dynamics of EA into a CN and then into a CML. CN could be analyzed and can give us some information about the dynamics of the EA, e.g., we can investigate for example centralities (Kantarci and Labatut, 2013), to figured out which vertex (i.e. an individual in the population) is most important in the algorithm. Also, we can investigate diameter, clustering coefficient or some advanced properties, as described in (Boccaletti, Latora, Moreno, Chavez and Hwang, 2006), (Holme and Saramäki, 2012). According to these, we can improve the algorithm to be better and to be able to find the optimal solution faster.

Both systems, i.e. CN and CML they can be controlled by unconventional methods as for example EAs are, (Zelinka, Celikovsky and Chen, 2010), (Zelinka, 2009). Thus a possibility how EA can control EA is there and was already reported in (Zelinka, Senkerik and Navrátil, 2006), (Zelinka et al., 2010). In this paper, we discuss the next step, i.e. control of EA dynamics via its CML visualization. The possibility to control classical (i.e. chaos based) CML by EAs has been already reported in various papers. Different kind of behavior can be then observed and analyzed in such CML as well as used in control (Schöll and Schuster, 2008). The whole process is symbolically depicted in Fig. 1.

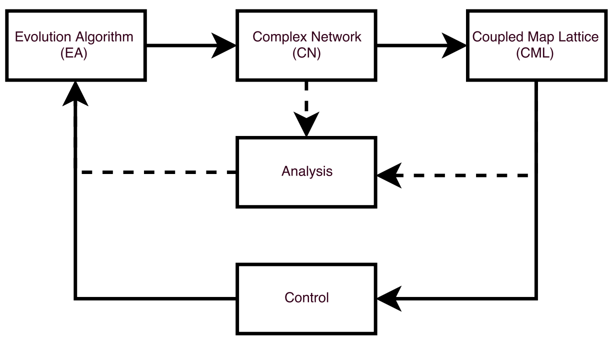

Figure 1. Motivation

In whole three kinds of experiments are reported here. The differential evolution (DE), firstly presented by

Price and Storn in 1995 (Storn and Price, 1995), has been selected for all of them. DE belongs to the class of algorithms called evolution algorithms (EA). DE with other algorithms like firefly algorithm (Yang, 2009), genetic algorithm (Holland, 1992), particle swarm optimization (Kennedy, 2011), self-organizing migrating algorithm (Zelinka, 2004), (Davendra, Zelinka et al., 2016) create a powerful tool for solving many optimization problems (Price, Storn and Lampinen, 2006), including control of complex systems.

The primary goal of this article, based on our previous research (Zelinka, 2015), is to show, how we can convert EA dynamics, mainly DE, into CN. Then conversion of the CN into CML is discussed and demonstrated, a) how behavior of the CML (i.e. EA) change when pinning inputs applied and b) how can be modified the dynamic of EA through CML interaction and feedback loop.

2 Experiment Design

2.1 Used Evolutionary algorithms

For experiments has been used differential evolution (Price et al., 2006) that is a population-based optimization method that works on real-number-coded individuals and can be regarded like discrete dynamical system. Its brief description (Price et al., 2006) is as follows. EA works with population, that consist of *N* possible solutions, called individuals. For each individual $\vec{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\vec{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ to a randomly selected third individual $\vec{x}_{r3,G}$. The resulting individual $\vec{x}'_{i,G}$ is crossed-over with the original individual $\vec{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\vec{u}_{i,G+1}$, is then compared with the fitness of $\vec{x}_{i,G}$. If the fitness of $\vec{u}_{i,G+1}$ is greater than the fitness of $\vec{x}_{i,G}$, then $\vec{x}_{i,G}$ is replaced with $\vec{u}_{i,G+1}$; otherwise, $\vec{x}_{i,G}$ remains in the population as $\vec{x}_{i,G+1}$. DE is quite robust, fast, and effective, with a global optimization ability. It does not require the objective function be differentiable, and it works well even with noisy, epistatic and time-dependent objective functions. For more about DE see (Price et al., 2006).

2.2 Parameters and cost function

During the experiments, we set DE (Storn and Price, 1995) parameters like this: $D = 50$, $NP = 30$, $CR = 0.8$, $F = 0.9$, $Generations = 1000$, $SearchingSpace = [-10^4, 10^4]^D$. To initialize DE dynamics and keep it active for longer time, Rastrigin function (1) has been selected for our experiments. The use of this function (as well as another test cost functions) and algorithm setting guarantee that DE will not converge quickly and thus its dynamics can be observed and recorded longer time.

$$10D\sum_{i=1}^{D} x_i^2 - 10\cos\left(2\pi x_i\right) \qquad (1)$$

2.3 Complex network design

EAs work with population, i.e. set of vectors that are called individuals, that are under mutual interactions. It can be recorded like network (social, complex,...). In CN then edges represent interactions between individuals, vertices represent individuals in a population, and vertices properties give us information about individuals (i.e. quality, fitness, ...), e.g., DE with 50 individuals can be represented as CN with 50 vertices, and degree property of node i tell us, how many individuals individual interacted with individual i. CN can be, for example, created from DE dynamics so that in DE-Rand1Bin for the creation of noise vector are required three individuals i_1, i_2 and i_3. With them and selected individual i a trial vector is created. If cost value of this vector is better than cost value of individual i this individual is replaced with the trial vector and three edges with weight equals to one from vertex i_1, i_2 and i_3 are added to vertex i. If the edge exists we just raise its weight, Fig. 2 that finally lead to the emergence of network, as visible in Fig. 4.

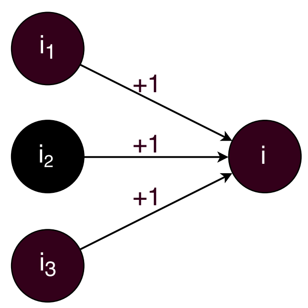

Figure 2. Conversion DE to CN

In such model, edges only raising its weight, but in many real-world networks connection between two vertices getting weaker, e.g., when for example ants travel they are releasing pheromones and thus they are creating roads. The pheromones are vaporized in the time. Their roads create a network and in this network edges are roads between two places and these roads are as stronger as how many ants go over them. So if ants stop walking over one road, this road disappear. In our model is forgetting realized as removing 5 percent of each weight after each 10th generation.

This is not the only model that can be use. We can create CN differently or for other algorithm. Some ideas were presented in (Pluhacek, Senkerik, Viktorin and Zelinka, 2016), (Zelinka, 2016), (Zelinka et al., 2016).

2.4 Used complex network properties

In this article, all experiments are based on the property called OutStrength. OutStrength of selected vertex is given as the sum of edges weights outgoing from this vertex. We can count OutStrength of vertex i according to equation (2) where w_{ij} is edge weight between vertex i and vertex j, and V represents number of vertices.

$$OutStrength_i = \sum_{j,i \neq j}^{V} w_{ij}. \quad (2)$$

2.5 CML design

CN based on EA dynamics could be analyzed (Skanderova and Fabian, 2015), (Davendra, Zelinka, Metlicka, Senkerik and Pluhacek, 2014), (Senkerik et al., 2016), or can be converted into CML. In the CML the row can be understand the property of CN (in our case OutStrength) or and this is another approach, row reflect time evolution of the CN edge. To create CML from CN is quite simple. As an interaction between vertices - individuals in evolution (i.e. weights of edges) is ϵ connection transferring pining between vertices (rows in CML) while vertex behavior of CN in time is in fact captured like row in CML. Comparing to the CML from school books (Schöll and Schuster, 2008), this CML is much more complex, because ϵ connections change in time its strength as well as joined vertices, see Fig. 3. Beside this philosophy, CN as CML can be visualized so that each row represent one edge in the CN, Fig. 5. Both approaches has been used here.

On Fig. 5 the CLM of all edges for DE in version/strategy DERand1Bin is depicted. Changes on edges during the time, during all generations, are visible. Each line represents one edge, so each line represents interactions between two selected individuals. On lines 1 to 9, we can see edges from vertex number 1 to all other vertices. On lines 11 to 19, we can see edges from vertex 2 to all vertices. Black lines are separators, and self-loops are not included. On this figure of CML weights are represented by color. Maximal weight is around the value of 10 and it is represented by yellow color, and we can observe some yellow dots during the algorithm run. These dots represent many positive interactions between individuals connected by this edge in a short time. Zero weight is represented by black color, and we can see many zero weights between generations 4000 and 5000. During these generations, we create only a few new better individuals.

Edges conversion into CML is better for a lower number of individuals. With a bigger number of individuals number of possible interactions growth, and a number of lines in CML growth too. As already suggested, we can convert edges into lines of CML, or each line of CML can represent selected property. For a bigger number of individuals, we select the second option, as we can see on Fig. 6. On this figure, we can see OutStrength CML of all vertices during the whole algorithm for Rastrigin function. Strength is represented by a color. Maximal strength is around value of 20 and it is represented by yellow color. Zero value is represented by black color. We can see that maximal strength reach individuals at the beginning of the algorithm. Then after few generations, strength is for all individuals mostly under the value of 10, and only some individuals reach higher strength for several generations.

3 Results

3.1 Experiment 1

In order to control EA dynamics (via its CML), it is important to realize, what the feedback and what influence EA dynamics is. CML is only the result of EA and it only visualizes the progress of some property of EA via CN. One possibility is to select individuals with probability according to CML state and values. Individuals in DE in next generations will be selected still randomly, **but** each with the different probability (based on CML values). We visualize in CML OutStrengt, so we will select individuals according to this property. The individual with bigger strength will be selected more often than the individual with lower strength. It can be stated, that individual i will be selected with probability p according to equation (3). $Prop$ represents some CN property, in our case OutStrength.

$$p_i = \frac{Prop_i}{\sum_{j,i \neq j}^{V} Prop_j} \quad (3)$$

If there is no edge in the graph, we will not be able to select individuals according to this equation. Also if some individuals reach zero strength, we will not able to select them, so we add constant c to the equation, to ensure, that each individual has at least small chance to be selected (4).

$$p_i = \frac{c + Prop_i}{c \cdot NP + \sum_{j,i \neq j}^{V} Prop_j} \quad (4)$$

During our experiments, for demonstration, we set up $c = 1$, but we can select any other number. If we select huge number, this version of DE will behave like classical DE. If we select lower number, DE will select individuals according to CML, but with 0 value, we can reach state, where we will not be ale to select individuals.

Selection of the individuals with probability feedback is created, and DE dynamics change in the next generations according to its CML. CML of such version we can see in Fig. 7. According to the classical version of DE, strength reaches values of 100 for some individuals. Maximal value for classical version were 20, so some individuals reach 5 times bigger strength. In Fig.

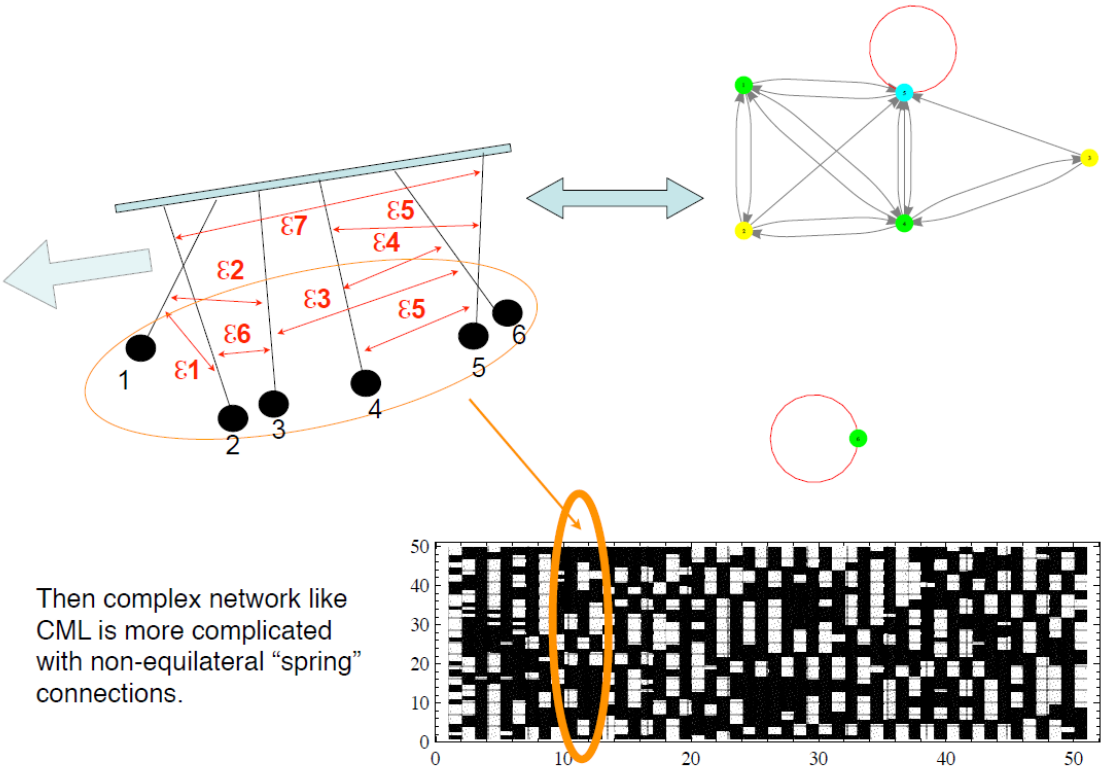

Figure 3. Principal conversion from CN to CML via pendulum analogy. An interaction between vertices (individuals in evolution) is ϵ connection transferring pining, vertex behavior of CN = row in CML.

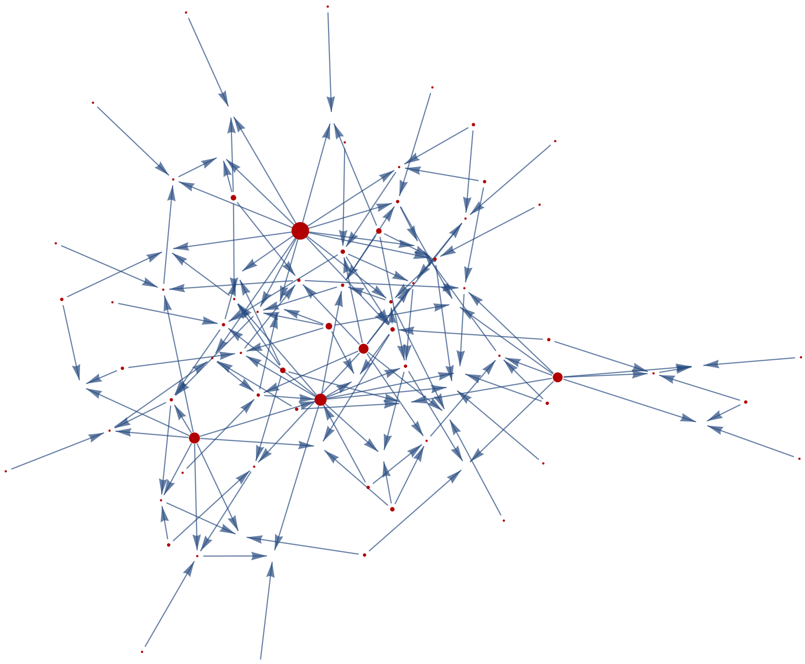

Figure 4. An example of network based on DERand1Bin dynamics

8 we can see same CML, but the range is changed. All values higher than 20 are represented by yellow color. The difference between the uncontrolled DE and DE with feedback is now more visible, see Fig. 6. It is visible that individuals reach much higher strength on version of the algorithm with feedback. Many individuals have strength higher than 20 during the algorithm for a long time and for many generations.

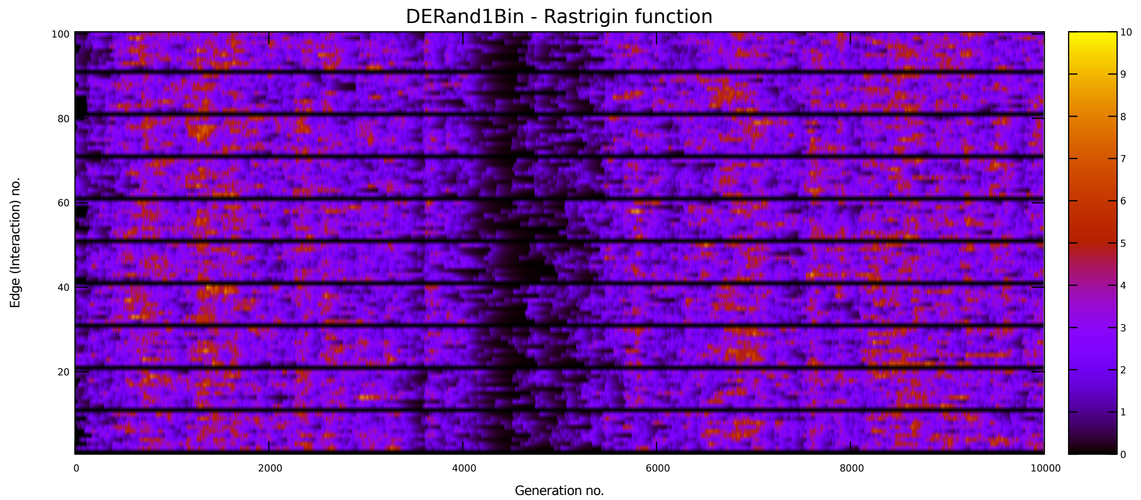

Figure 5. CML of all edges (interactions between individuals) for DERand1Bin

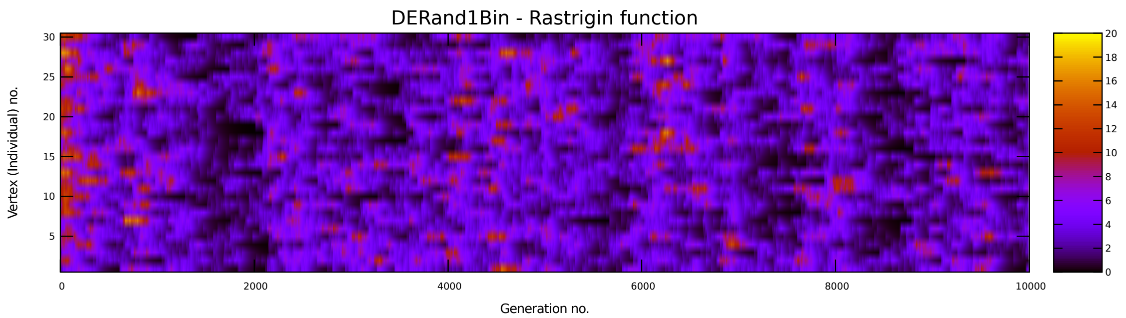

Figure 6. OutStrength CML of all vertices (individuals) for DERand1Bin

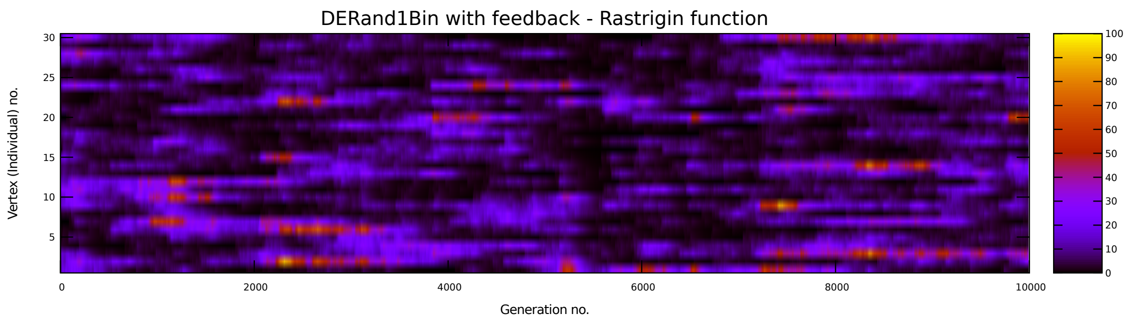

Figure 7. OutStrength CML of all vertices (individuals) for DERand1Bin with feedback

3.2 Experiment 2

The second experiment was focused on question whether DE can be controlled through CML similarly as in (Zelinka et al., 2010). CML can be also classically controlled as was described in (Schöll and Schuster, 2008). In that case of CML control suitable control inputs, so-called pinning values, has to be calculated. Each individual i.e. each line of CML can be affected by pinning input. This input changes the probability of selecting an affected individual, and the algorithm changes its behavior. With added pinning we can select individuals according to equation (5). $Ping_i$ is control input for individual i. Other variables are same as in previous equations.

$$p_i = \frac{c + Prop_i \cdot Ping_i}{c \cdot NP + \sum_{j,i \neq j}^{V} Prop_j \cdot Ping_j} \qquad (5)$$

With these pinning inputs we can affect the algorithm. On Fig. 9 is captured CML of DE with pings that has been periodically changed between sites. After each 500 generations a ping has been changed ran-

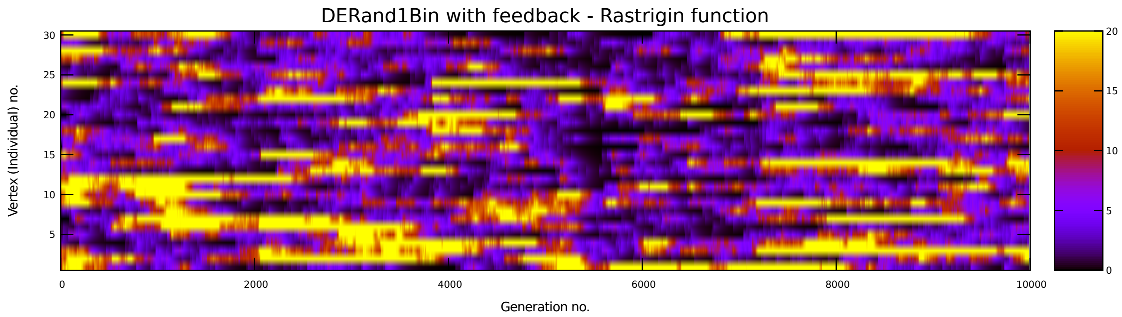

Figure 8. OutStrength CML of all vertices (individuals) for DERand1Bin with feedback - smaller range

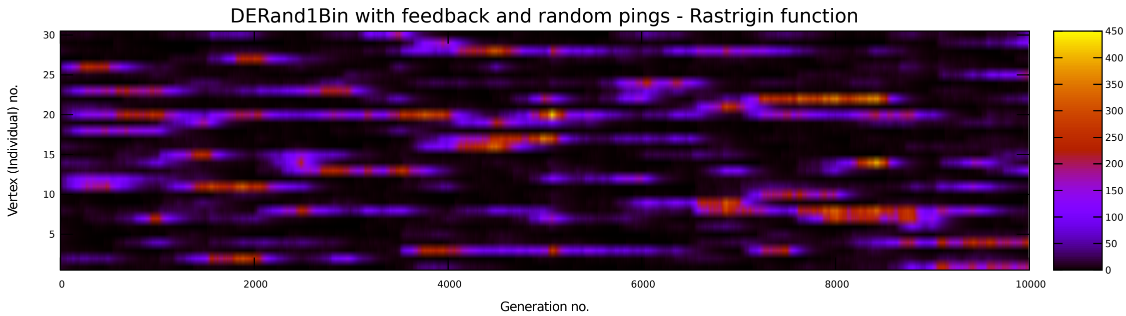

Figure 9. CML of DERand1Bin with feedback and random pings

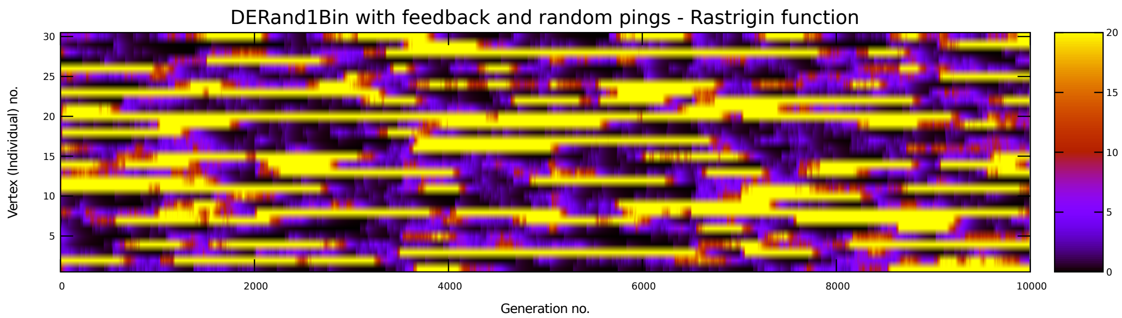

Figure 10. CML of DERand1Bin with feedback and random pings - smaller range

domly in the interval $[0, 3]$, thus, the dynamic of DE also changed and the maximal strength raise again. The maximal value of strength is now 450, so much bigger. In more close look, we can see how algorithm change its behavior after change of pings. For better comparison the range of this CML has been changed, so all values higher that 20 are represented by yellow color. This CML with change range as visible on Fig. 10.

3.3 Experiment 3

To show, that the dynamic of algorithm change, the last simple experiment was done. In this experiment, individuals were divided into two groups, and one group was excited by strong ping = 3 and the second group inhibited by weak ping = 0.1. After each 500 generations the ping between those two groups has been exchanged. Result of this experiment is on Fig. 11. We can see, that ping really changes the algorithm dynamics. CML behavior was divided into two groups, one group with strong pings and one group with weak pings, after each 500 generations, when the pings changed, the groups behavior also change.

4 Conclusion

In this paper, is reported progress on the research of three areas evolution algorithms, complex network, and CML. Possibility on control of EAs dynamics via its CN and CML visualization was sketched and three simple experiments were performed in order to show that CML derived from EAs dynamics behave in the same way as the classical CML and thus it shall be controllable, even by unconventional methods as reported in (Zelinka, 2009) or in (Schöll and Schuster, 2008).

Conversion of EA into a CN and to the CML has been

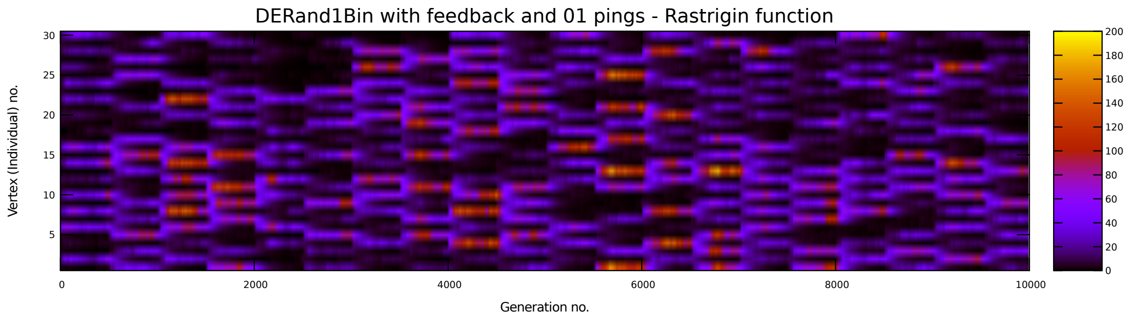

Figure 11. CML of DERand1Bin with feedback and pings on two groups

revised and reported. The control of DE, which select individuals according to its CML has been discussed. This gives us the opportunity to control DE (generally any EA), that can be controlled through CML by its control inputs - pinning sites. We also showed how such CML looks like on selected property called OutStrength in each version of DE (classical DE, DE with feedback, DE with random pings and DE with pings on two groups).

Based on this, CML can be created according to many other CN properties like degree, closeness centrality, clustering coefficient and many others. Also different models of CN and different algorithms can be used. An investigation on how various type of pining can affect the algorithm and its performance can be further done. Not only for CML with OutStrength, but also for other properties as well.

The future research will be focused on full control of EA dynamics in the same manner as in (Zelinka, 2009), (Schöll and Schuster, 2008).

Acknowledgment

The following grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/15/06700S and by Grant of SGS No. SP2017/134, VSB Technical University of Ostrava.

References

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. and Hwang, D.-U. (2006), 'Complex networks: Structure and dynamics', *Physics reports* **424**(4), 175–308.

Davendra, D., Zelinka, I., Metlicka, M., Senkerik, R. and Pluhacek, M. (2014), Complex network analysis of differential evolution algorithm applied to flowshop with no-wait problem, *in* 'Differential Evolution (SDE), 2014 IEEE Symposium on', IEEE, pp. 1–8.

Davendra, D., Zelinka, I. et al. (2016), 'Self-organizing migrating algorithm', *New Optimization Techniques in Engineering* .

Holland, J. H. (1992), *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press.

Holme, P. and Saramäki, J. (2012), 'Temporal networks', *Physics reports* **519**(3), 97–125.

Kantarci, B. and Labatut, V. (2013), Classification of complex networks based on topological properties, *in* 'Cloud and Green Computing (CGC), 2013 Third International Conference on', IEEE, pp. 297–304.

Kennedy, J. (2011), Particle swarm optimization, *in* 'Encyclopedia of machine learning', Springer, pp. 760–766.

Pluhacek, M., Senkerik, R., Viktorin, A. and Zelinka, I. (2016), Creating complex networks using multiswarm pso, *in* 'Intelligent Networking and Collaborative Systems (INCoS), 2016 International Conference on', IEEE, pp. 180–185.

Price, K., Storn, R. M. and Lampinen, J. A. (2006), *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media.

Schöll, E. and Schuster, H. G. (2008), *Handbook of chaos control*, John Wiley & Sons.

Senkerik, R., Viktorin, A., Pluhacek, M., Janostik, J. and Oplatkova, Z. K. (2016), Study on the time development of complex network for metaheuristic, *in* 'Artificial Intelligence Perspectives in Intelligent Systems', Springer, pp. 525–533.

Skanderova, L. and Fabian, T. (2015), 'Differential evolution dynamics analysis by complex networks', *Soft Computing* pp. 1–15.

Storn, R. and Price, K. (1995), 'Differential evolution–a simple and efficient adaptive scheme for global optimization over continuous spaces, berkeley', *CA: International Computer Science Institute* .

Yang, X.-S. (2009), Firefly algorithms for multimodal optimization, *in* 'International symposium on stochastic algorithms', Springer, pp. 169–178.

Zelinka, I. (2004), Somaself-organizing migrating algorithm, *in* 'New optimization techniques in engineering', Springer, pp. 167–217.

Zelinka, I. (2009), 'Real-time deterministic chaos control by means of selected evolutionary techniques', *Engineering Applications of Artificial Intelligence* **22**(2), 283–297.

Zelinka, I. (2015), 'A survey on evolutionary algorithms dynamics and its complexity–mutual relations, past, present and future', *Swarm and Evolutionary Computation* **25**, 2–14.

Zelinka, I. (2016), 'On mutual relations amongst evolutionary algorithm dynamics and its hidden complex network structures: An overview and recent advances', *Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications* p. 215.

Zelinka, I., Celikovsky, S. and Chen, G. (2010), *Evolutionary algorithms and chaotic systems*, Springer.

Zelinka, I., Senkerik, R. and Navrátil, E. (2006), 'Investigation on realtime deterministic chaos control by means of evolutionary algorithms', *IFAC Proceedings Volumes* **39**(8), 190–196.

Zelinka, I., Tomaszek, L. and Kojecky, L. (2016), On evolutionary dynamics modeled by ant algorithm, *in* 'Intelligent Networking and Collaborative Systems (INCoS), 2016 International Conference on', IEEE, pp. 193–198.