

# Investigation on Relations Between Complex Networks and Evolutionary Algorithm Dynamics

Ivan Zelinka<sup>1</sup>, Donald Davendra<sup>2</sup>

<sup>1,2</sup> Department of Informatics  
Faculty of Electrical Engineering and Computing Science,  
Technical University of Ostrava, Tr. 17. Listopadu 15,  
Ostrava, Czech Republic  
*ivan.zelinka@vsb.cz, donald.davendra@vsb.cz*

**Abstract:** In this article we discuss relations between the so-called complex networks and dynamics of evolutionary algorithms. The main aim of this article is to investigate whether it is possible to model (or visualize) evolutionary dynamics as complex networks, whose connections will represent interactions amongst the individuals during all generations. Our simulations are based on selected evolutionary algorithms (2 algorithms in 10 versions) and 16 test functions. Data obtained through the simulations were processed graphically as well as statistically.

**Keywords:** evolutionary dynamics, complex networks, differential evolution, SOMA.

## I. Introduction

In this article, we try to merge two completely different (at first glance) areas of research: complex networks and evolutionary computation.

Large-scale networks, exhibiting complex patterns of interaction amongst vertices exist in both nature and in man-made systems (i.e., communication networks, genetic pathways, ecological or economical networks, social networks, networks of various scientific collaboration, Internet, World Wide Web, power grid etc.). The structure of complex networks thus can be observed in many systems.

The word “complex” networks [1], [2] comes from the fact that they exhibit substantial and non-trivial topological features, with patterns of connection between vertices that are neither purely regular nor purely random. Such features include a heavy tail in the degree distribution, a high clustering coefficient, hierarchical structure, amongst other features. In the case of directed networks, these features also include reciprocity, triad significance profile and other features.

Amongst many studies, two well-known and much studied classes of complex networks are the scale-free networks and small-world networks (see examples in Figures 1 and 2), whose discovery and definition are vitally important in the scope of this research. Specific structural features can be observed in both classes i.e. so called power-law degree distributions for the scale-free networks and short path lengths with high clustering for the small-world networks.

Research in the field of complex networks has joined together researchers from many areas, which were outside of this interdisciplinary research in the past like mathematics, physics, biology, chemistry computer science, epidemiology etc...

Evolutionary computation is a sub-discipline of computer science belonging to the “bio-inspired” computing area. Since the end of the second world war, the main ideas of evolutionary computation has been published [3] and widely introduced to the scientific community [4]. Hence, the “golden era” of evolutionary techniques began, when Genetic Algorithms (GA) by J. Holland [4], Evolutionary Strategies (ES), by Schwefel [5] and Rechenberg [6] and Evolutionary Programming (EP) by Fogel [7] had been introduced. All these designs were favored by the forthcoming of more powerful and more easily programmable computers, so that for the first time interesting problems could be tackled and evolutionary computation started to compete with and became a serious alternative to other optimization methods.

The main idea of our research is to show in this article that the dynamics of evolutionary algorithms, in general, shows properties of complex networks and evolutionary dynamics can be analyzed and visualized like a complex network. This article is focused on observation and description of complex networks phenomenon in evolutionary dynamics. Possibilities of its use are discussed at the end.

## II. Motivation and Preliminaries

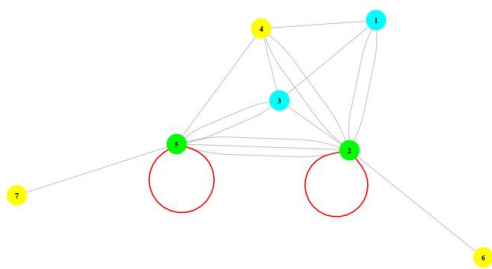
Motivation of this research is quite simple. As mentioned in the introduction, evolutionary algorithms (EA) are capable of hard problem solving. A number of examples on evolutionary algorithms can easily be found. EAs use with chaotic systems has been done for example in [8] where EAs has been used on local optimization of chaos, [9] for chaos control with use of the multi-objective cost function or in [10, 11] where evolutionary algorithms have been studied on chaotic landscapes. Slightly different approach with evolutionary algorithms is presented in [12] where selected algorithms were used to synthesize artificial chaotic systems.

In [13], [14] EAs has been successfully used for real-time chaos control and in [15] EAs was used for optimization of Chaos Control.

Other examples of evolutionary algorithms application can be found in [16], which developed statistically robust evolutionary algorithms, alongside research conducted by [17]. Parameters of permanent magnet synchronous motors has been optimized by PSO and experimentally validated on the servomotor. Another research was focused on swarm intelligence, which has been used for IIR filter synthesis, co-evolutionary particle swarm optimization (CoPSO) approach for the design of constrained engineering problems, particularly for pressure vessel, compression spring and welded beam, etc. Another research base on EAs has been done in [18] where EAs has been used for fast and accurate watermark retrieval or in [19] - Differential Evolution has been analysed on variants on unconstrained global optimization problems.

On the other side, complex networks, widely studied across many branches of science are promising and is a modern interdisciplinary research. EA's based on its canonical central dogma (following Darwinian ideas) clearly demonstrate intensive interaction amongst individual in the population, which is, in general, one of the important attributes of complex networks (intensive interaction amongst the vertices).

The main motivation (as well as a question) is whether it is possible to visualize and simulate underlying dynamics of an evolutionary process like a complex network. The reason for this undertaking is because various techniques for analysis and control of complex networks currently exist and if a complex network structure would be hidden behind EA dynamics, then we believe, that for example the above-mentioned control techniques could be used to improve the dynamics of EAs. All experiments here were designed to analyse and either confirm or reject this idea.



**Figure 1.** Example of a small network.

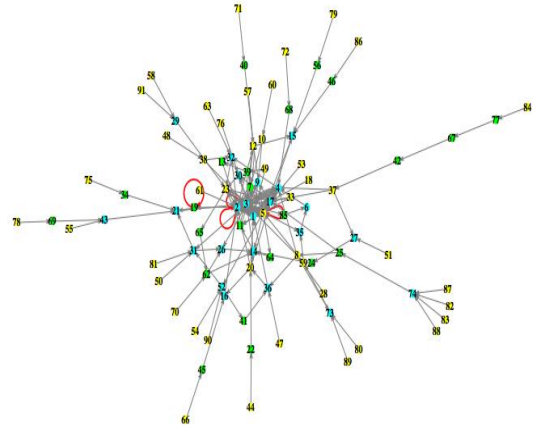
### III. Experiment Design

#### A. Selected algorithms and its settings

For the experiments described here, stochastic optimization algorithms, such as Differential Evolution (DE) [20] and Self Organizing Migrating Algorithm (SOMA) [21], have been used. Application of alternative algorithms like GA and Simulated Annealing (SA), ES and/or Swarm Intelligence are now in process.

All experiments have been done on a special server consisting of 16 Apple XServer (2 x 2 GHz Intel Xeon, 1 GB

RAM), each with 4 CPU, so in total 64 CPUs were available for calculations. It is important to note here, that such technology was used to save time due to a large number of calculations, however it must be stated that evolutionary identification described here, is also solvable on a single PC (with longer execution time). For all calculations and data processing, Mathematica version 7.0.1.0 was used.



**Figure 2.** Example of a more complex network with multiple edges and selfloops.

Four versions of SOMA and six versions of DE have been utilized for all experimentations in this paper. See Table 1 – Table 4 for relation between each version and index corresponding to other Tables. Parameters for the optimizing algorithm were set up in such a way as to reach similar value of maximal cost function evaluations for all used versions. Each version of EAs has been applied 50 times in order to get less or more valuable statistical data.

DE is a population-based optimization method that works on real-number-coded 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  $\vec{x}_{r_1,G}$  and  $\vec{x}_{r_2,G}$  to a randomly selected third individual  $\vec{x}_{r_3,G}$ . 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 global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy, epistatic and time-dependent objective functions. The DE schema is depicted in Figure 23.

SOMA is a stochastic optimization algorithm, modelled on the social behavior of competitive-cooperating individuals [21]. It was chosen because it has been proved that this algorithm has the ability to converge towards the global optimum [21]. SOMA works on a population of candidate solutions in loops, called migration loops. The

population is initialized by being randomly and uniformly distributed over the search space at the beginning of the search. In each loop, the population is evaluated and the solution with the lowest cost value becomes the Leader. Apart from the Leader, in one migration loop, all individuals will traverse the searched space in the direction of the leader. Mutation, the random perturbation of individuals, is an important operation for evolutionary strategies. It ensures the diversity among all the individuals and it also provides a means to restore lost information in a population. Mutation is different in SOMA as compared with other evolutionary strategies. SOMA uses a parameter called PRT to achieve perturbations. This parameter has the same effect for SOMA as mutation for GA. The novelty of this approach lies in that the PRT vector is created before an individual starts its journey over the search space. The PRT vector defines the final movement of an active individual in the search space. The randomly generated binary perturbation vector controls the permissible dimensions for an individual. If an element of the perturbation vector is set to zero, then the individual is not allowed to change its position in the corresponding dimension. An individual will travel over a certain distance (called the PathLength) towards the Leader in a finite number of steps in the defined length. If the PathLength is chosen to be greater than one, then the individual will overshoot the Leader. This path is perturbed randomly. The schema of SOMA is depicted at Figure 24.

The primary aim here is not to show which version of EA is better or worse, but to show that dynamics of the EAs can in reality be described and analyzed as complex networks.

Algorithm	Index
AllToOne	S1
AllToRandom	S2
AllToAll	S3
AllToAllAdaptive	S4

Table 1. Used Versions of SOMA.

Algorithm	Index
DEBest1JIter	D1
DEBest2Bin	D2
DELocalToBest	D3
DERand1Bin	D4
DERand1DIter	D5
DERand2Bin	D6

Table 2. Used Versions of DE.

	S1	S2	S3	S4
PathLength	3	3	3	3
Step	0.11	0.11	0.11	0.11
PRT	0.1	0.1	0.1	0.1
PopSize	50	50	50	50
Migrations	300	300	300	300
MinDiv	-1	-1	-1	-1
Individual Length	50	50	50	50

Table 3. Parameter settings of SOMA.

	D1 – D6
NP	50
F	0.9
CR	0.3
Generations	3000
Individual Length	50

Table 4. Parameter settings of DE.

### B. Selected test functions and its dimensionality

The test functions applied in this experimentation were selected from the test bed of 17 test functions. In total 16 test function were selected as a representative subset of functions which shows geometrical simplicity and low complexity as well as functions from the “opposite side of spectra”. Selected functions (see Figure 3) were: 1st DeJong (1), Schwefel’s function (6), Rastrigin’s function (5), Ackley’s function (10) amongst others (see Equations (1)-(16)). Each of them has been used for identification of complex networks dynamics and structure in 50 dimensions (individual length was 50). Dimension is in the formulas (1)-(16) represented by variable D, so as one can see, it is easy to calculate selected functions for an arbitrary dimension. Functions (1)-(16) has been selected due to their various complexity and mainly for the fact that this functions are widely used by researchers working with evolutionary algorithms. Another reason was that the speed of convergence and thus evolutionary dynamics itself, is different for simple functions like (1) or more complex example (13).

### C. Data for complex network visualization

The most critical point of this research and related simulations was as to which data and relations should be selected and consequently visualized. Based on the investigated algorithms, we believe that there is no universal approach, but rather a “personal” one, based on the knowledge of algorithm principle. Of course, some conclusions (see section Conclusion) can be generalized over a class or family of algorithms. As mentioned in the previous sections, algorithms like DE and SOMA were used. Each class of algorithm is based on a different principle. The main idea was such that each individual is represented by vertex and edges between vertices should reflect dynamics in population, i.e. interactions between individuals (which individual has been successfully used for offspring creation,...).

$$\sum_{i=1}^D x_i^2 \quad (1)$$

$$\sum_{i=1}^{D-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2 \quad (2)$$

$$\sum_{i=1}^D |x_i| \quad (3)$$

$$\sum_{i=1}^D i x_i^4 \quad (4)$$

$$2D \sum_{i=1}^D x_i^2 - 10 \cos(2\pi x_i) \quad (5)$$

$$\sum_{i=1}^D -x_i \sin(\sqrt{|x_i|}) \quad (6)$$

$$1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (7)$$

$$-\sum_{i=1}^{D-1} \left( 0,5 + \frac{\sin(x_i^2 + x_{i+1}^2 - 0,5)^2}{(1 + 0,001(x_i^2 + x_{i+1}^2))^2} \right) \quad (8)$$

$$\sum_{i=1}^{D-1} \left( 4\sqrt{(x_i^2 + x_{i+1}^2)} \sin(50 \sqrt[10]{(x_i^2 + x_{i+1}^2)^2 + 1}) \right) \quad (9)$$

$$\sum_{i=1}^{D-1} \left( \frac{1}{e^5} \sqrt{(x_i^2 + x_{i+1}^2)} + 3(\cos(2x_i) + \sin(2x_{i+1})) \right) \quad (10)$$

$$\sum_{i=1}^{D-1} \left( 20 + e - \frac{20}{e^{0,2\sqrt{\frac{(x_i^2 + x_{i+1}^2)}{2}}}} - e^{0,5(\cos(2\pi x_i) + \cos(2\pi x_{i+1}))} \right) \quad (11)$$

$$\sum_{i=1}^{D-1} \left( -x_i \sin(\sqrt{|x_i - x_{i+1} - 47|}) - \left( x_{i+1} + 47 \right) \sin\left(\sqrt{\left| x_{i+1} + 47 + \frac{x_i}{2} \right|} \right) \right) \quad (12)$$

$$\sum_{i=1}^{D-1} \left( x_i \sin(\sqrt{|x_{i+1} + 1 - x_i|}) \cos(\sqrt{|x_{i+1} + 1 + x_i|}) + \left( x_{i+1} + 1 \right) \cos(\sqrt{|x_{i+1} + 1 - x_i|}) \sin(\sqrt{|x_{i+1} + 1 + x_i|}) \right) \quad (13)$$

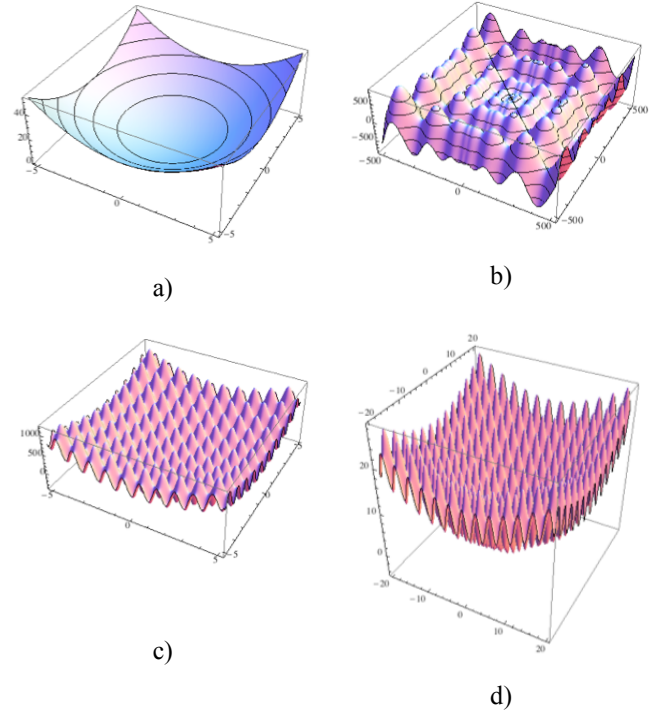
$$\sum_{i=1}^{D-1} \left( 0,5 + \frac{\sin(\sqrt{100x_i^2 - x_{i+1}^2} - 0,5)}{(1 + 0,001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2))} \right) \quad (14)$$

$$\sum_{i=1}^{D-1} \left( -1 \left( \sin(x_i) \sin\left(\frac{x_i^2}{\pi}\right)^{20} + \sin(x_{i+1}) \sin\left(\frac{2x_i^2}{\pi}\right)^{20} \right) \right) \quad (15)$$

$$\sum_{i=1}^{D-1} \left( e^{\frac{-(x_i^2 + x_{i+1}^2 + 0,5x_i x_{i+1})}{8}} \cos(4\sqrt{x_i^2 + x_{i+1}^2 + 0,5x_i x_{i+1}}) \right) \quad (16)$$

The SOMA algorithm, as described in [21], consists of a Leader attracting the entire population in each migration loop (equivalent of generation), so in that class of swarm-like algorithm, it is clear that the position in the population of activated Leaders shall be recorded as vertex (getting new inputs from remaining vertices - individuals) and used (with remaining part of population) for visualization and statistical data processing. The other case is DE, e.g. DERand1Bin in which each individual is selected in each generation to be a parent. Thus in DE, we have recorded only those individuals-parents, that has been replaced by better offspring (like vertex with added connections). In the DE

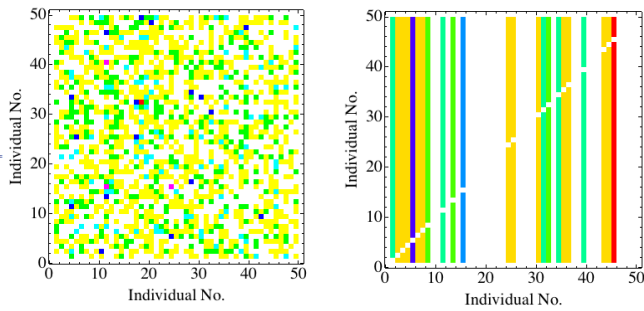
class of algorithms we have omitted the philosophy that a bad parent is replaced by a better offspring, but accepted the philosophical interpretation, that the individual (worse parent) is moving to the better position (better offspring). Thus no vertex (individual) has to be either destroyed or replaced in the philosophical point of view. If, for example, DERand1Bin has a parent been replaced by offspring, then it was considered as an activation (new additional links, edges) of vertex-worse parent from three another vertices (randomly selected individuals, see [20]).



**Figure 3.** Figure 3. Selected test functions: 1st DeJong (a), Schwefel's function (b) Rastrigin's function (c) and Ackley's function (d).

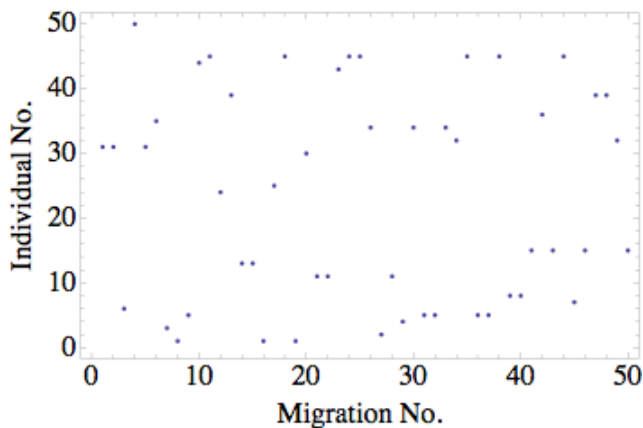
#### D. Visualization methods

Experimental data can be visualized in a few different ways and as an example, a few typical visualizations is depicted here. For example in Figure 4 interactions between individuals in the population during entire evolution is described. As mentioned in the previous section, vertices in complex graph are individuals that are activated by other individuals, incrementally from generation to generation. This can be visualized as in Figure 4, where DE and SOMA population examples are depicted. Different colors represent different number of inputs to vertex (different activations of selected individual). White color represent no relations (activations) between individuals, e.g. white square between individual 6 (y axe) and 3 (x axe) means that individual 6 was never used to compete for the position in the new population or to create new offsprings with individual 2. Philosophy of competition or offspring creation is based on the principles of the used algorithm (see previous subsection).



**Figure 4.** Table of interaction, DE (left) and SOMA (right). On both axes are individuals. White color means no interaction between individuals. See SOMA (right) where diagonal white line means that the Leader individual cannot select himself. From SOMA interactions is clear that some newer individuals have been selected (white columns).

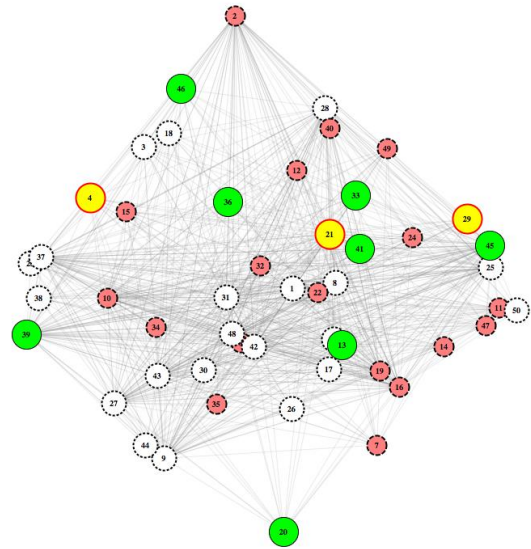
Another kind of visualization is depicted in Figure 5, in which one can see which individual (out of 50) has been activated for offspring creation (in this case selected like Leader in SOMA). Information from Figure 5 is in close relation with Figure 4 – white columns from Figure 4. (unused individuals – never selected for Leaders) are empty rows (without dots) as shown in Figure 5.



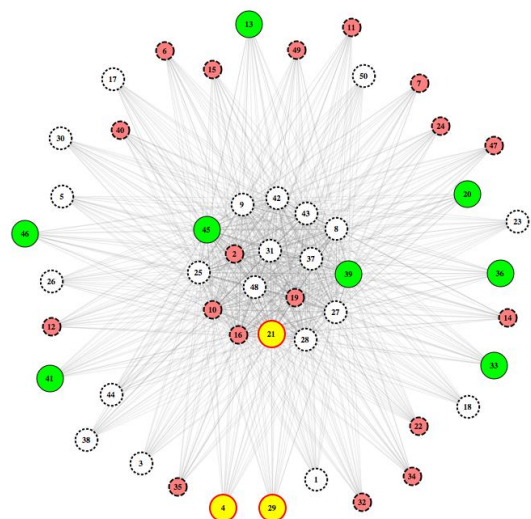
**Figure 5.** An example of activated leaders (y axis) with dependence on Migrations (x axis) of SOMA. For example 1st individual (y axis) has been selected 3 times during 50 Migrations (x axis).

Figures 4 and 5 are sort of auxiliary visualizations, which does not give total view on complex network structure behind evolutionary dynamics. Better visualization that can be used is as in Figure 6, 7 or 8, which shows, that interactions between individuals create (at the first glance) structures, which looks like complex networks. However, it has to be said, that we have met results whose visualizations looks like net and resemble complex networks but after closer complex network characteristics calculations, those networks did not belong to the class of complex networks with small world phenomenon. Meaning of vertices in the above mentioned figures is given by ratio of incoming and outgoing edges and implies that: small vertex (small gray (pink) with dashed edges) has less incoming edges than outgoing. White (middle-sized) vertex is balanced (i.e. has the same incoming number of edges as outgoing) and dark gray (green), the biggest, are vertices with more incoming edges than outgoing. The light gray (yellow) vertex is the

most activated individual – vertex with the maximum of incoming edges. In EA jargon, small vertex is an individual, which has been used more times for offspring creation rather than as a successful parent and pink vertices reflects the opposite.

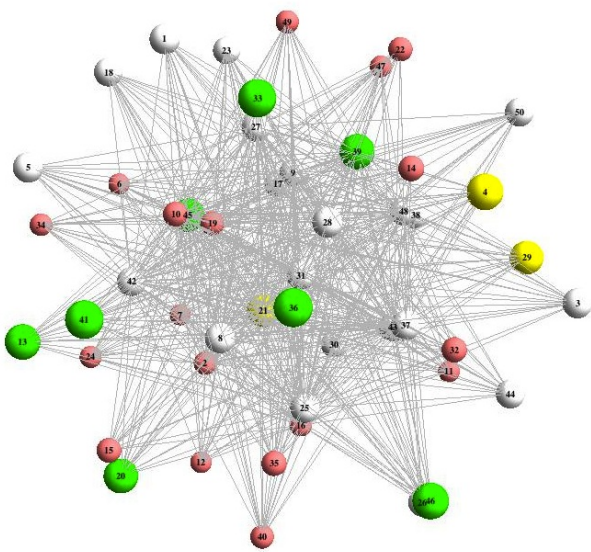


**Figure 6.** Complex network example of SOMA dynamics in a “natural” format. Colors of vertices has the following meaning, as described in the text: small vertex (small gray (pink) with dashed edges) has less incoming edges than outgoing. White (middle-sized) vertex is balanced (i.e. has the same incoming number of edges as outgoing) and dark gray (green), the biggest, are vertices with more incoming edges than outgoing. The light gray (yellow) vertex is the most activated individual – vertex with the maximum of incoming edges. In EA jargon, a small vertex is an individual, which has been used more times for offspring creation rather than as a successful parent and pink vertices reflects the opposite



**Figure 7.** The same network as in Figure 6 in a

“symmetrical” format.

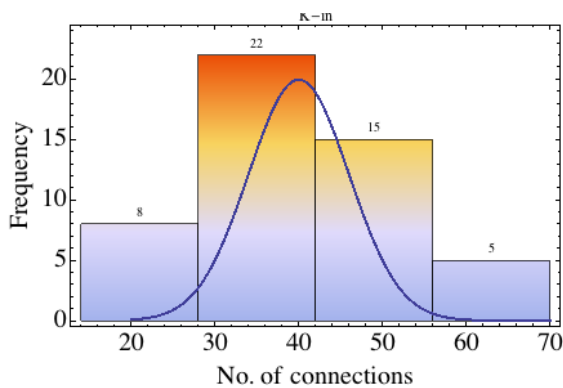


**Figure 8.** 3D vizualization of complex network depicted in Figure 6.

To ensure that an algorithm and its dynamics actually investigated for complex network phenomenon can be really understood and modelled like a complex network, typical characteristics has been calculated, for example the distribution of vertices degree (see Figure 10).

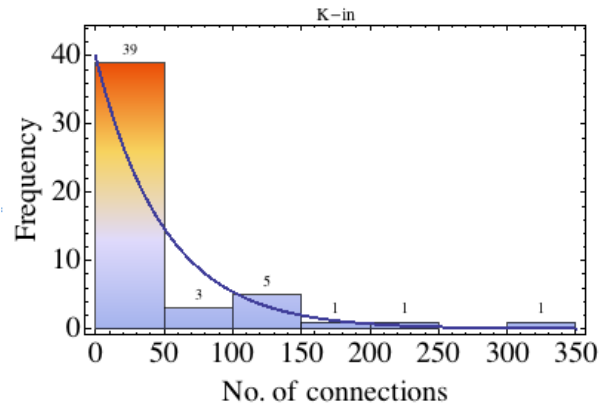
**IV. Results**

As reported above, both algorithms, in 10 versions, has been tested on various test function (to reveal its complex networks dynamics) with constant level of test function dimensionality (i.e. individual length) and different number of generations (migrations) in all used algorithms. All data has been processed graphically (e.g. Figure 10-12, etc.) alongside calculations of basic statistical properties. Emergence of complex network structure behind evolutionary dynamics depend on many factors, however some special versions of used algorithms did not show complex network structure despite the fact that the number of generations was quite large. All main ideas coming from the results are discussed in the next subsection.

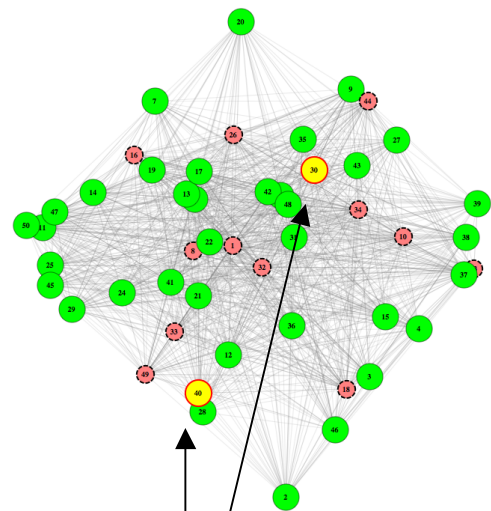


**Figure 9.** An example of DERand1Bin exhibiting normal-like distribution of vertices degree, no complex networks has been observed behind the evolutionary

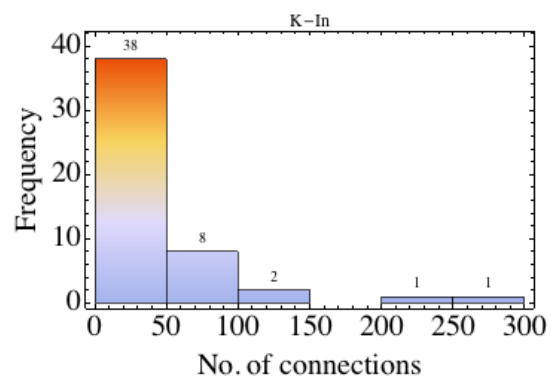
dynamics.



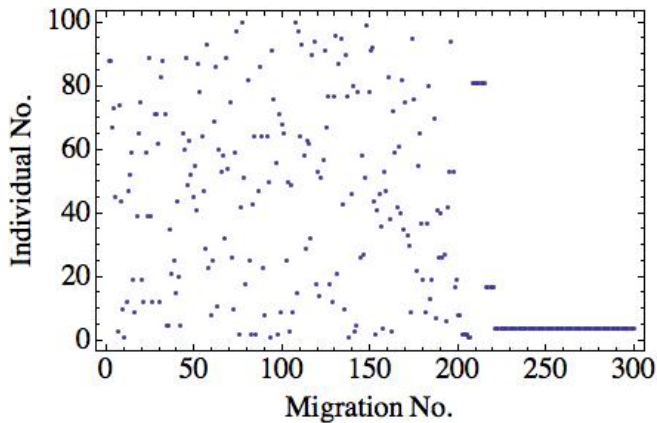
**Figure 10.** An example of histogram exhibiting long tail distribution of vertices degree, typical result for SOMA swarmlike algorithm. Basically only one individual / vertice has more than 350 in-coming connections, while most of them (39) have less than 50. Vizualization can of course be „reversed“, i.e. if on the y axis would be No. of connections as opposed to the x axis, then similar distribution, typical for complex networks analysis and visualization, would be generated.



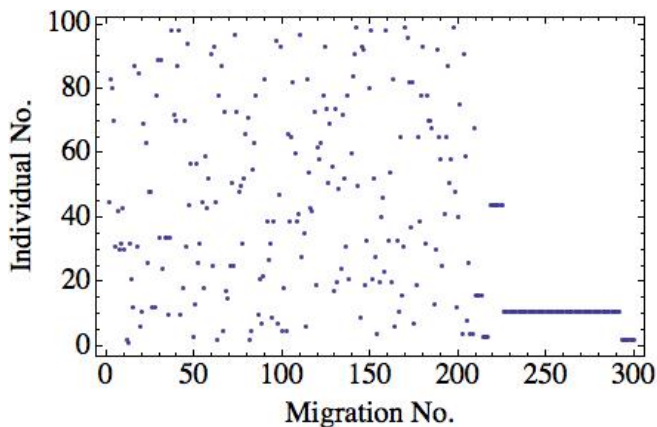
**Figure 11.** Complex network of the DELocalToBest with two of the most intensively connected vertices (individuals)...



**Figure 12.** ... and its histogram of the vertices connections (note that two vertices are with almost 300 connections each).



**Figure 13.** An example of activated Leaders at the moment when evolution has found the global extreme. In such a moment the best individual is repeatedly selected (see line after 230 migrations) and become to be extremely attractive node of all.



**Figure 14.** An example of activated leaders at the moment when evolution has found global extreme.

## V. Conclusion

The main motivation of this research is whether it is possible to visualize and simulate underlying dynamics of an evolutionary process as a complex network. Based on **preliminary** results (based only on 2 algorithms in 10 versions and 16 test function out of 17) it can be stated that:

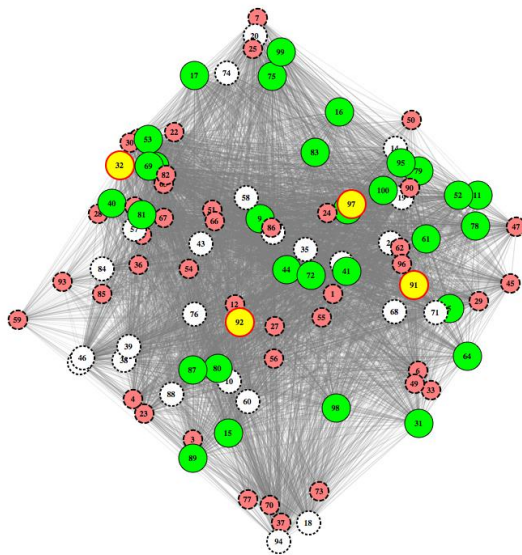
- **No. of generations:** occurrence of the complex network structure (CNS) sensitively depends on the number of generations. If the number of generations was small, then no CNS was established. This effect can be easily understand so that low number of generations means that EAs has not long enough “time“, to establish CNS. This is quite a logical observation in complex network dynamic when CNS is not observable at the beginning of linking process. During our experiments, it has been observed that the “moment“ of CNS establishment depends on the cost function dimension, population size, used algorithm and cost function. Generally, EAs searching for global extreme is at the beginning, quite random-like and when the “domain“ of global extreme is discovered, then CNS is quite quickly established.
- **Dimensionality:** impact on CNS forming has been observed when the dimension of the cost function was big and number of generations was too low, The selected EA was not able to successfully finish the global extreme

search – not all connections had been properly established. Thus if high dimensional cost functions are used, then the number of generations has to be selected so that at least the “domain“ of the global extreme is found. On the other side, if the number of generations (or Migrations in the case of the SOMA algorithm) is very large, then it is possible to observe the effect of “rich becoming richer“, i.e. one vertex (individual) becomes the winner repeatedly, see Figure 13 and 14. This moment usually means that global extreme has been found and further searching is not necessary.

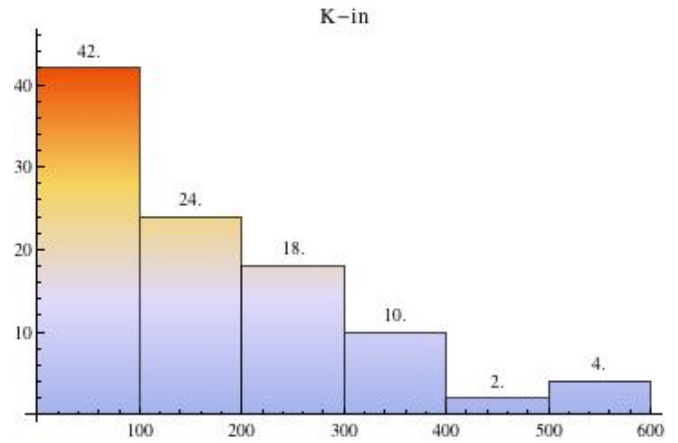
- **Test functions:** dependence of CNS forming on the test function was not strictly observed, the general consensus being that for more complex test functions, like Schwefel (6), etc, the algorithm needs more generations to establish CNS, i.e. more complex function requires more generations and/or bigger population size. In the case of simpler functions like 1<sup>st</sup> De Jong (1) and low dimensions, the global extreme is quickly found and phase of CNS creation is very short which epitomizes the last phase “rich become to be richer“ (see Figure 13 and 14), as mentioned in the previous paragraph. It is important to say that this last phase depends on the algorithm structure and that we have observed it in the case of SOMA (S1, S2) algorithm and DE (D3). The term “algorithm structure” means that for example in the case of SOMA, the Leader (winning vertex) selected as the first individual of initial population with the best fitness, no matter how many other individuals in the population has the same fitness. This concept is demonstrated in Figures 15 – 22. On these figures are depicted CNS dynamics of SOMA applied on Schwefel’s function (6) in dimension 50, with 100 individuals and 300 migrations. These figures depict just the forming of CNS (based on data visualized in Figure 16) with histogram depicted in Figure 17. In this phase, the network has four equal vertices. Different situations arose when the global extreme is found; for example when the first individual with the best fitness is selected to be a best solution (vertex) it automatically get incoming links from remaining members of the population (see Figures 18-20). If evolution continues further, then the winner becomes to be more preferred (still getting more and more incoming connections). It is visible in Figure 21 and mainly in Figure 22.
- **Population size:** CNS forming was observed usually from population size of 20 and more individuals for dimensions 50. Again, it is the parameter, which does not influence CNS forming alone, but in a combination with another parameters, as mentioned in the previous items.
- **Used algorithm:** CNS forming has also been clearly observed with algorithms, that are more or less based on swarm philosophy or partly associated with it. For example DERand1Bin did not show any CNS forming (in principle each individual is selected to be a parent), see Figure 9, while in the case of the DELocalToBest (Figure 12) in which the best solution in the population play an important role, CNS has been observed, as well as in the SOMA strategies (see Figure 11). The conclusion reached is that CNS forming is more likely observable with swarm like algorithms rather than “randomly removed” algorithms. We think that this is quite logical and close to the idea of preferred linking in the complex networks modelling social behavior (citation networks, etc...)
- **Possible use:** based on numerically demonstrated fact (no mathematical proof has been made) that EAs

dynamics can be visualized like complex networks we believe that there is new research area for study of EAs dynamics and its possible control via techniques of complex network control [23]. Also another domain of research is to study information flow in such a network by means of so called k-shell decomposition.

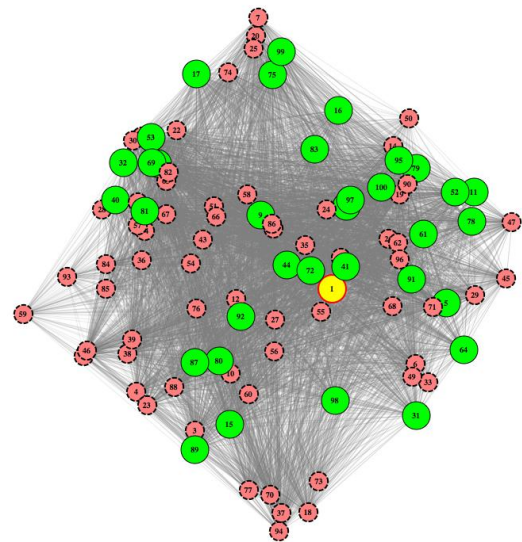
Results presented here are an extension of [22] and shall be taken into consideration as “preliminary”, because more robust and statistically massive simulations are needed for conclusions, which are currently in process. However, based on the results reported here, it can be stated, in general, that the dynamics of evolutionary algorithms can be visualized as complex network structures and therefore the features of complex networks can probably be used to measure the efficiency of the used algorithm, or to control its dynamics in the future research.



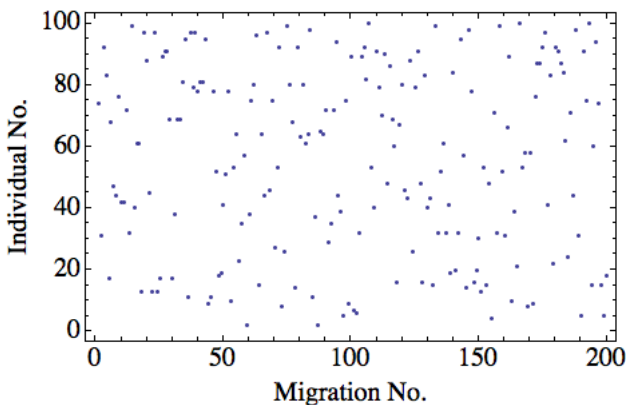
**Figure 15.** CNS related to SOMA after 200 migrations (see Figure 16). The number of the best vertices (with maximal number of connections) is now 4.



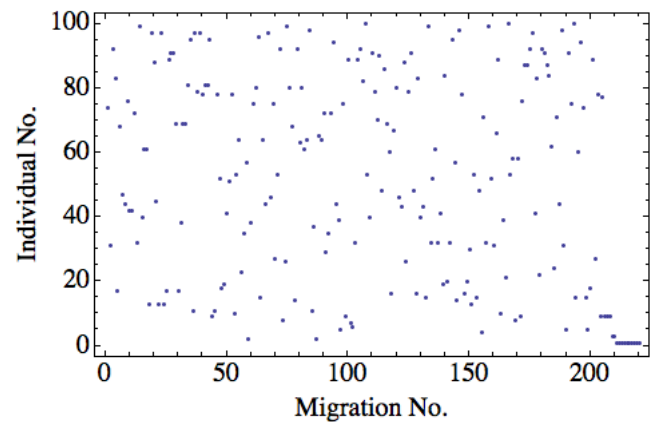
**Figure 17.** Histogram of the vertices connections based on Figure 15.



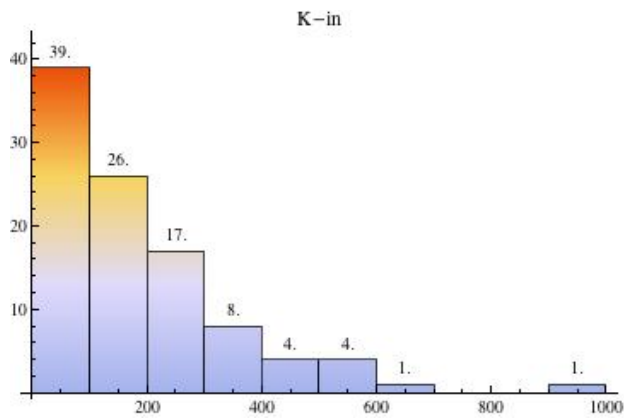
**Figure 18.** CNS related to SOMA after 220 migrations (see Figure 19). The number of the best vertices (with maximal number of connections) is now only 1 – winner take all.



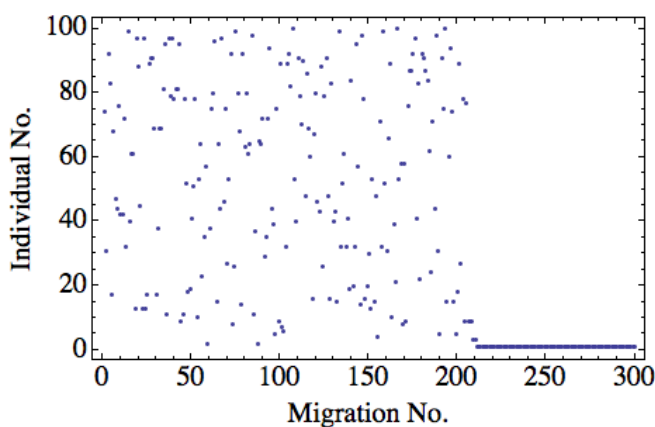
**Figure 16.** Activated leaders (y axe) with dependence on Migrations (x axe) SOMA.



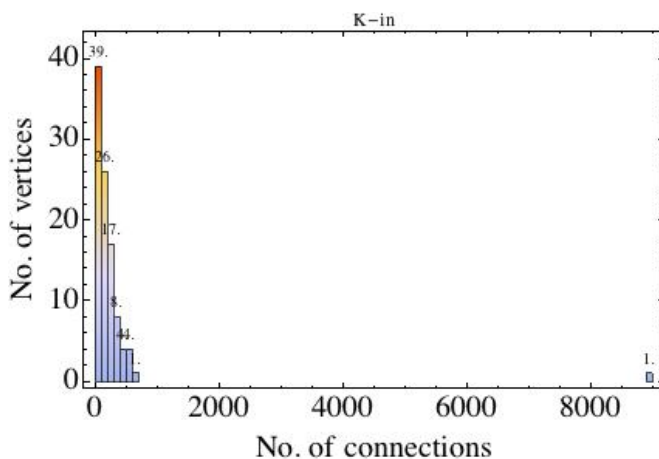
**Figure 19.** Activated leaders (y axe) with dependence on Migrations (x axe) SOMA. Note that between 200 – 220 individuals, the winner is the first individual in the population.



**Figure 20.** Histogram of the vertices connections based on Figure 19. The first individual is now the most “rich“ vertex with 780 connections.



**Figure 21.** Activated leaders (y axe) with dependance on Migrations (x axe) SOMA. Note that between 200 – 300 individuals, the winner is still the first individual in the population.



**Figure 22.** Histogram of the vertices connections based on Figure 21. Thanks to the fact that the first individual is still selected as the winner, it get a lot of connetions (8700) and histogram basically does not looks like typical long tail distribution.

## Acknowledgment

This work was supported by grant No. MSM 7088352101 of the Ministry of Education of the Czech Republic and by grants of the Grant Agency of the Czech Republic GACR 102/09/1680.

## References

- [1] S. N. Dorogovtsev and J.F.F. Mendes, Evolution of Networks, Adv. Phys. 51, 1079 (2002)
- [2] S. Boccaletti et al., Complex Networks: Structure and Dynamics, Phys. Rep., 424 (2006), 175-308.
- [3] Turing, A.: Intelligent machinery, unpublished report for National Physical Laboratory. In: Michie, D. (ed.) Machine Intelligence, vol. 7 (1969); Turing, A.M. (ed.): The Collected Works, vol. 3, Ince D. North-Holland, Amsterdam (1992)
- [4] Holland, J.: Adaptation in natural and artificial systems. Univ. of Michigan Press, Ann Arbor (1975)
- [5] Schwefel, H.: Numerische Optimierung von Computer-Modellen, PhD thesis (1974); Reprinted by Birkhauser (1977)
- [6] Rechenberg, I.: (1971) Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (PhD thesis), Printed in Fromman-Holzboog (1973)
- [7] Fogel, D.B.: Unearthing a Fossil from the History of Evolutionary Computation. Fundamenta Informaticae 35(1-4), 1-16 (1998)
- [8] Richter H. and Reinschke K. J., (2000), Optimization of local control of chaos by an evolutionary algorithm, Physica D, 144, pp. 309-334
- [9] Richter H., An evolutionary algorithm for controlling chaos: The use of multi-objective fitness functions, in Parallel Problem Solving from Nature-PPSN VII (Eds.: Guervós M., Panagiotis J. J., Beyer A., Villacanas F. H. G., J. L. & Schwefel H. P.), Lecture Notes in Computer Science, Vol. 2439 (Springer-Verlag, Berlin), (2002), pp. 308-317
- [10] Richter H., Evolutionary Optimization in Spatio-temporal Fitness Landscapes, Lecture Notes In Computer Science, Springer-Verlag, NUMB 4193, pp. 1-10, (2006) ISSN 0302-9743
- [11] Richter H., A study of dynamic severity in chaotic fitness landscapes, Evolutionary Computation, 2005. The IEEE Congress, Volume 3, Issue 2-5 Sept. 2005, (2005), pp. 2824 – 2831, Vol. 3
- [12] Zelinka I., Chen G., Celikovskiy S., Chaos Synthesis by Means of Evolutionary Algorithms, International Journal of Bifurcation and Chaos, University of California, Berkeley USA, Vol 18, No 4, (2008), p. 911 – 942
- [13] Zelinka, I., Real-time deterministic chaos control by means of selected evolutionary algorithms, Engineering Applications of Artificial Intelligence, (2008), doi:10.1016/j.engappai.2008.07.008
- [14] Zelinka I, Investigation on Realtime Deterministic Chaos Control by Means of Evolutionary Algorithms, In: 1st IFAC Conference on Analysis and Control of Chaotic Systems, Reims, France, (2006)
- [15] Senkerik R., Zelinka I., and Navratil E., Optimization of feedback control of chaos by evolutionary algorithms, In: : 1st IFAC Conference on Analysis and Control of Chaotic Systems, Reims, France, (2006)
- [16] Dashora, Y., et al., Improved and generalized learning strategies for dynamically fast and statistically robust evolutionary algorithms. Engineering Applications of

Artificial Intelligence, (2007),  
doi:10.1016/j.engappai.2007.06.005

- [17] Li L., Wenxin L., and David A. C., (2007), Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors, *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2007.10.002
- [18] G. Jeyakumar, C. Shunmuga Velayutham, An Empirical Performance Analysis of Differential Evolution Variants on Unconstrained Global Optimization Problems, In: *International Journal of Computer Information Systems and Industrial Management Applications*, Vol. 2, pp. 77–86, 2010
- [19] E.V.Gopal, Munaga.V.N.K.Prasad, V.Ravi, Fast and Accurate Watermark Retrieval Using Evolutionary Algorithms In: *International Journal of Computer Information Systems and Industrial Management Applications*, Vol. 2, pp. 121–136, 2010
- [20] Price, K. *An Introduction to Differential Evolution*, *New Ideas in Optimization*, eds. Corne, D., Dorigo, M. & Glover, F. (McGraw-Hill, London, UK), pp. 79–108. (1999)
- [21] Zelinka I., (2004), SOMA – Self Organizing Migrating Algorithm“,Chapter 7, 33 p. In: B.V. Babu, G. Onwubolu (eds), *New Optimization Techniques in Engineering*, Springer-Verlag, ISBN 3-540-20167X
- [22] Zelinka I., Davendra D., Snášel V., Jašek R., Šenkeřík R., Oplatková Z., Preliminary Investigation on Relations Between Complex Networks and Evolutionary Algorithms Dynamics, *CISIM 2010*, Poland
- [23] Sean Meyn, *Control Techniques for Complex Networks*, Cambridge University Press, 2007

## Author Biography



**Ivan Zelinka** (ivanzelinka.eu) was born in Czech Republic, and went to the Technical University of Brno, where he studied technical cybernetics and obtained his degree in 1995. He obtained his Ph.D. degree in Technical Cybernetics in 2001 at Tomas Bata University in Zlin, He is now a Professor at the Technical University in Ostrava, Czech Republic. His specialization is artificial intelligence and its interdisciplinary use and applications. His e-mail address is: ivan.zelinka.vsb.cz.



**Donald Davendra's** research background is in the fields of Evolutionary Algorithms, Chaotic Systems and their application to combinatorial optimization problems. He has a Ph.D. in Technical Cybernetics from Tomas Bata University in Zlin. Currently he holds the position of Assistant Professor at the Faculty of Electrical Engineering and Computing Science at the Technical University of Ostrava, Czech Republic.

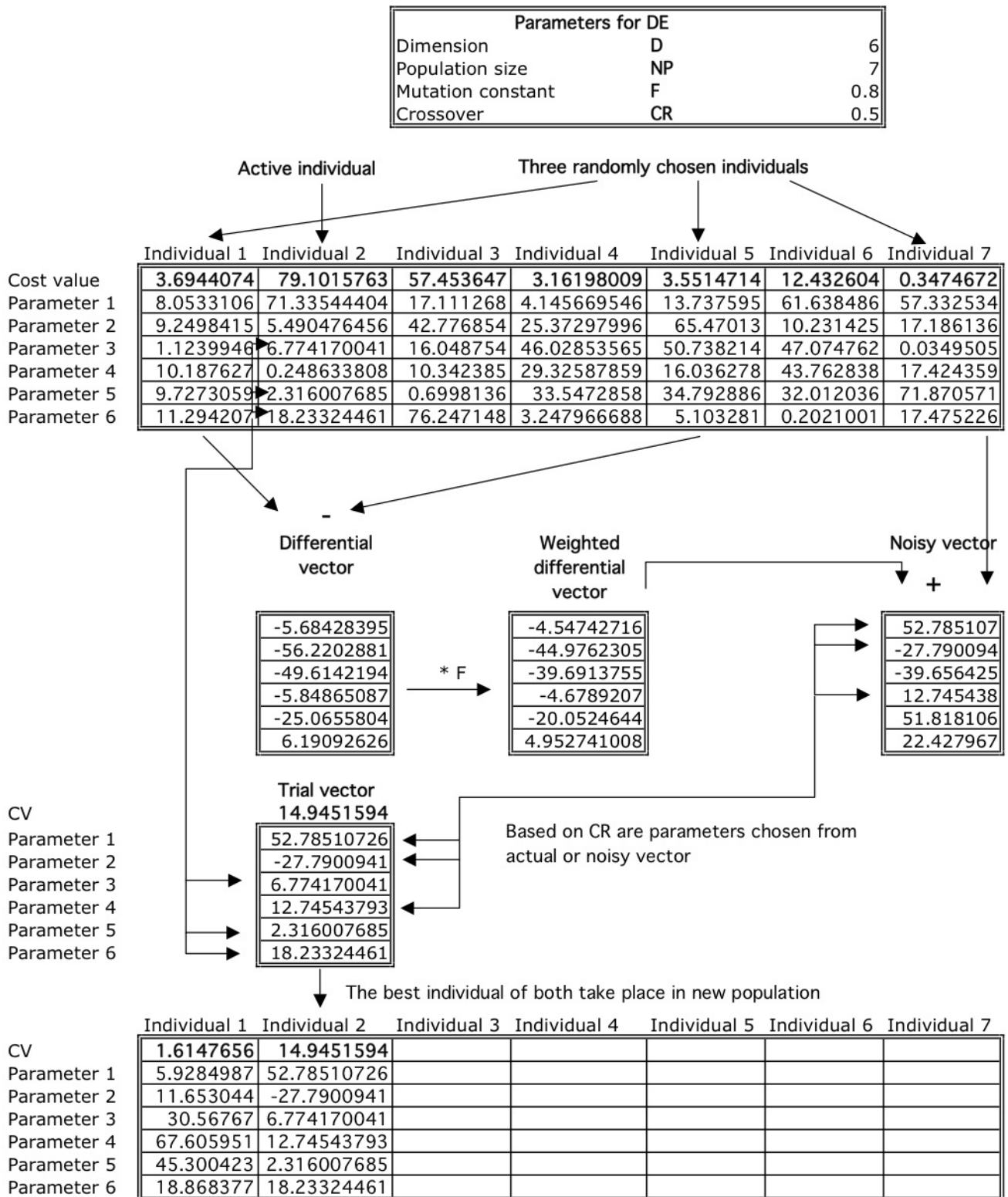


Figure 23. Schema of Differential Evolution.

Control parameter		PRT vector	
Step	0.3	If Rand < PRT then 1 else 0	↔
Mass	3	If Rand < PRT then 1 else 0	↔
PRT	0.1	If Rand < PRT then 1 else 0	↔
AcceptedError	0.1	If Rand < PRT then 1 else 0	↔
Migrations	1000	If Rand < PRT then 1 else 0	↔
NP	7	If Rand < PRT then 1 else 0	↔

Cost function  $f(x) = \text{Abs}(\text{Parameter } 1) + \text{Abs}(\text{Parameter } 2) + \dots + \text{Abs}(\text{Parameter } 6)$

	Travelling individual				Leader		
	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5	Individual 6	Individual 7
CostValue	204.91528	261.363199	163.79679	121.73019	107.52784	121.06024	120.20974
Parameter 1	3.0615753	-46.635691	5.0246553	38.723912	35.822343	0.0715185	23.761224
Parameter 2	2.5117282	54.0366854	85.104704	0.2928606	24.111443	4.2879691	20.384665
Parameter 3	46.75014	51.2828942	11.347164	3.0796963	24.657689	60.241731	33.437248
Parameter 4	72.486617	15.0801288	2.916686	3.6713463	5.8142407	4.5385164	4.0482021
Parameter 5	6.316564	57.1557444	58.829537	26.610056	12.43856	23.891907	4.2271271
Parameter 6	73.788657	-37.172056	0.5740442	49.352316	4.6835676	28.028598	34.351273

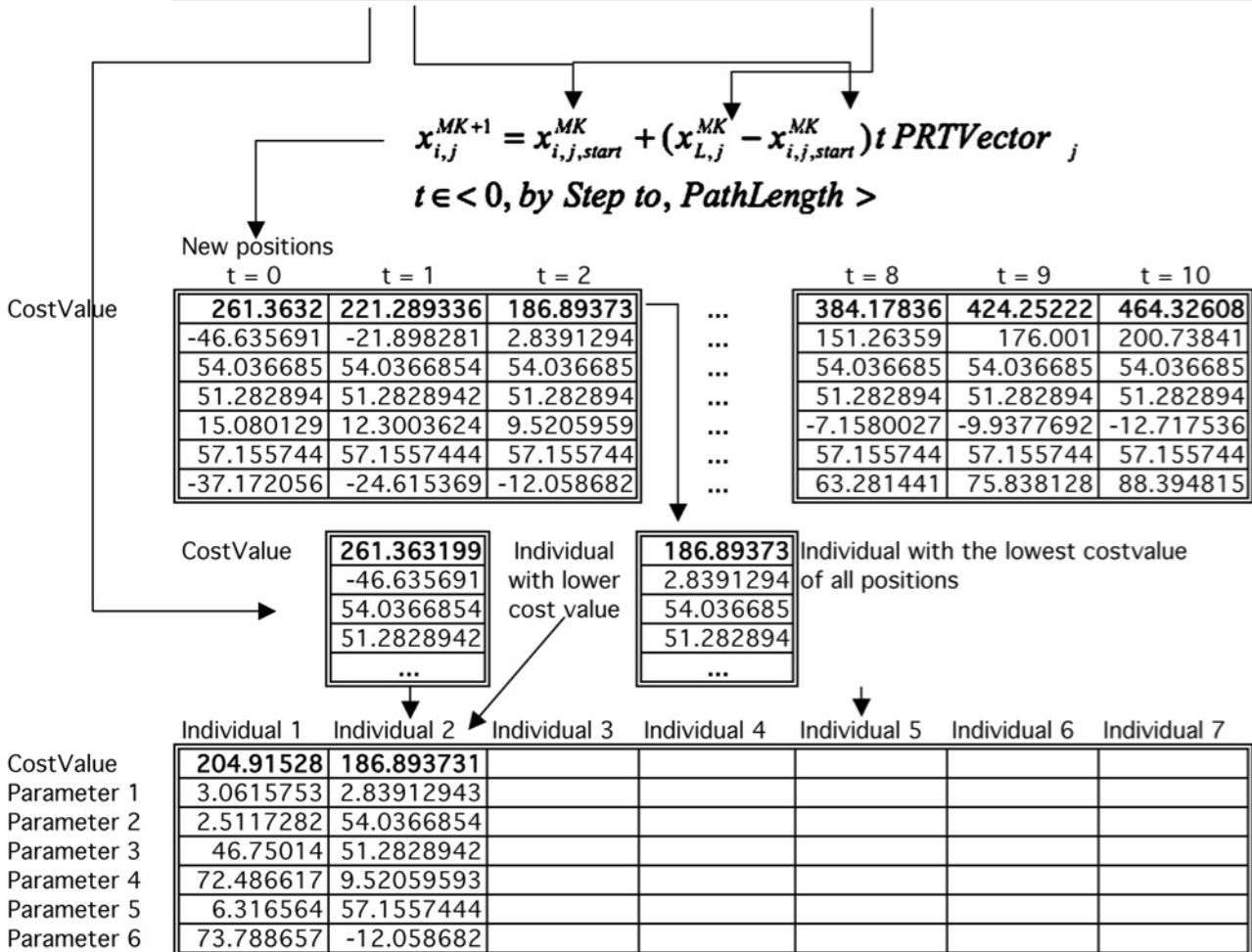


Figure 24. Schema of SOMA algorithm.