Do Evolutionary Algorithm Dynamics Create Complex Network Structures?

Ivan Zelinka*
Donald Davendra

Department of Informatics
Faculty of Electrical Engineering and Computer Science
Technical University of Ostrava, Tr. 17. Listopadu 15
Ostrava, 708 00, Czech Republic
**ivan.zelinka@vsb.cz*

Senkerik Roman
Jasek Roman

Faculty of Applied Informatics
University of Tomas Bata in Zlin, Nad Stranemi 4511
Zlin, 760 01, Czech Republic

This paper presents a novel method for visualizing the dynamics of evolutionary algorithms in the form of complex networks. We discuss the analogy between individuals in populations in an arbitrary evolutionary algorithm and vertices of a complex network, as well as between edges in a complex network and communication between individuals in a population. The possibility of visualizing the dynamics of a complex network using the coupled map lattices method and control by means of chaos control techniques are also discussed.

1. Introduction

In this paper, we try to merge two completely different (at first glance) areas of research: complex networks (CNs) and evolutionary algorithms (EAs). Large-scale networks, exhibiting complex patterns of interaction amongst vertices, exist in both natural and artificial systems (e.g., communication networks, genetic pathways, ecological or economical networks, social networks, networks of various scientific collaboration, the internet, World Wide Web, power grid, and others). The structure of CNs can be observed in many systems. The term "complex networks" [1, 2] comes from the fact that these networks exhibit substantial and nontrivial 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, and hierarchical structure, amongst others. In the case of directed networks, these features also include reciprocity, triad significance profile, and more. CNs are another field of research that can be linked with EAs. A special case is social networking, which is a complex, large, and expanding sector of

the information economy. Researchers' interest in this field is growing rapidly. It has been studied extensively since the beginning of the twentieth century. The first normative contributions in this area were proposed in the 1940s by sociologist M. Granovetter and mathematician L. C. Freeman. The basic theory, "The Strength of Weak Ties," was mentioned in 1973 [3]. Granovetter argued that within a social network, weak ties are more powerful than strong ties. Another significant principle was published in 1979 by L. C. Freeman [4]. This work presented the definition of centrality, which is one node's relationship to other nodes in the network. Freeman defined such basic metrics as degree, control, and independence that present researchers now use. Social network researchers have acquired data for their studies using various methods. In the past, these studies were only based on questionnaire data, which typically reached hundreds of individuals [5]. In the late 1990s, new technologies such as the internet and cellular phones enabled researchers to construct large-scale networks using emails [6], phone records [7], or web search engines [8].

Amongst many studies, two well-known classes of CNs are the scale-free and small-world networks, whose discovery and definition are vitally important in the scope of this research. Specific structural features can be observed in both classes, that is, 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 CNs has joined together researchers from many areas previously outside of this interdisciplinary research such as mathematics, physics, biology, chemistry, computer science, and epidemiology.

Evolutionary computation is a discipline of computer science belonging to the "bio-inspired" computing area. Since the end of World War II, the main ideas of evolutionary computation have been published [9] and widely introduced to the scientific community [10]. Hence, the "golden era" of evolutionary techniques began when genetic algorithms (GAs) by J. Holland [10], evolutionary strategies (ES) by Schwefel [11] and Rechenberg [12], and evolutionary programming (EP) by Fogel [13] had been introduced. All these designs were favored by the availability of powerful computers that are easy to program, 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.

2. Motivations and Ideas

The core idea of this research is to show that EA dynamics, in general, exhibit CN properties and that evolutionary dynamics can be analyzed and visualized as CNs. This idea is based on evaluating the EA dynamics (e.g., parent selection, offspring creation) under a few presumptions. The first is that each individual is considered to be a ver-

tex of the network. The second is that there are no parents and off-spring (as typically considered amongst evolutionists); each individual is considered to be alive and is not replaced by its offspring. A situation that is evaluated as being "replaced by offspring" is here decrypted as an individual moving from a worse to a better position in the space of possible solutions. The third and last very important presumption is that each individual that moves to a better position (i.e., in classical terminology "has been replaced by offspring") is in our philosophy a vertex that gets incoming edges from other parents (individuals, vertices) that have participated in this improvement. For example, in the case of differential evolution (DE), version DERand1Bin, usually four individuals (three randomly selected and one regularly selected) create new offspring (i.e., new position of an individual; remember it is a vertex). If this new offspring-vertex at the new position has better fitness, then it is a successful vertex that increases the number of incoming edges by three (randomly selected parents-vertices). If fitness is worse, then there are two possibilities. The first is that the number of incoming and outgoing edges will not be changed; the second is that each of three cooperating parents (vertices) will get one incoming edge from the final vertex (offspring) that has a worse fitness and its position will not be changed. In this way the EA dynamics can be easily transformed to a complex structure. This paper focuses on the observation and description of CN phenomena in evolutionary dynamics. Possibilities of its use are discussed in Section 5.

3. Evolutionary Algorithms and Complex Network Structures

For the experiments described here, stochastic optimization algorithms such as DE [14] and the self-organizing migrating algorithm (SOMA) [15] have been used. All experiments were performed on a special server consisting of 16 Apple XServers (2×2 GHz Intel Xeon, 1 GB RAM), each with four CPUs, so a total of 64 CPUs were available. It is important to note that such technology was used to save time due to the large number of calculations. However, it must be stated that the evolutionary identification described here can also be solved on a single personal computer with a longer execution time. *Mathematica* 7 was used for all calculations and data processing. Four versions of SOMA and two versions of DE were applied for all simulations in this paper. The optimizing algorithm parameters were set up in such a way as to reach a similar value of maximal cost function evaluations for all versions used. Each EA version was applied 50 and 100 times in order to get less or more valuable statistical data.

DE [14] 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 a global optimization ability. It does not require that the objective function be differentiable, and it works well even with noisy, epistatic, and time-dependent objective functions.

SOMA is a stochastic optimization algorithm based on the social behavior of competitive-cooperative individuals [15]. It was chosen because it has been proved that this algorithm has the ability to converge toward the global optimum [15]. 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 ES. It ensures diversity among all the individuals and also provides a means for restoring lost information in a population. Mutation is different in SOMA as compared with other ES. SOMA uses a parameter called PRT to achieve perturbations. This parameter has the same effect for SOMA as mutation for GAs. 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 path length) toward the leader in a finite number of steps in the defined length. If the path length is chosen to be greater than one, then the individual will overshoot the leader. This path is perturbed randomly.

▌ 3.1 Selected Test Functions and Dimensionality

The test functions applied in this experiment (in order to record the evolutionary dynamics induced by searching on those functions) were selected from a test bed of 17 test functions. In total, 16 test functions were selected as a representative subset of functions that shows geometrical simplicity and low complexity as well as functions from the

"opposite side of the spectrum." Selected functions include the first DeJong, Schwefel's function (equation (1)), Rastrigin's function, and Ackley's function (equation (5)) amongst others. Each of them was used for identifying the dynamics and structure of CNs in 50 dimensions (individual length was 50). Dimension is represented in the formulas by variable D, so as can be seen, it is easy to calculate selected functions for an arbitrary dimension. The functions mentioned were selected due to their variety of complexity and mainly for the fact that they are widely used by researchers working with EAs. Another reason was that the speed of convergence, and thus the evolutionary dynamics, are different for simple functions like the first DeJong or a more complex example like Rana's function.

$$\sum_{i=1}^{D} -x_i \sin\left(\sqrt{|x_i|}\right) \tag{1}$$

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

$$-\sum_{i=1}^{D-1} \left(0, 5 + \frac{\sin\left(x_i^2 + x_{i+1}^2 - 0, 5\right)^2}{\left(1 + 0, 001\left(x_i^2 + x_{i+1}^2\right)\right)^2}\right) \tag{3}$$

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

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

▌ 3.2 Data for Complex Network Visualization

The most critical point of this research and related simulations was determining 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 algorithmic principles. Of course, some conclusions can be generalized over a class or family of algorithms. As mentioned in the previous sections, algorithms such as DE and SOMA were used. Each class of algorithm is based on a different principle. The main idea is that each individual is represented by a vertex and edges between vertices should reflect population dynamics, that is, interactions between individuals (the individuals used to create off-

spring). The SOMA algorithm, as described in [15], consists of a leader attracting the entire population in each migration loop (equivalent of generation). In that class of swarm-like algorithm, it is clear that the position of activated leaders in the population can be recorded like a vertex (getting new inputs from remaining vertices-individuals) and used (with the remaining population) for visualization and statistical data processing. The other case is DE, for example, DERand1Bin in which each individual is selected in each generation to be a parent. Thus, in DE, we record only those individuals-parents that were replaced by better offspring (i.e., a 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 an individual (worse parent) is moving to a better position (better offspring). Thus, no vertex-individual has to be either destroyed or replaced in our philosophical viewpoint. If, for example, DERand1Bin has a parent that is replaced by an offspring, then it is considered as an activation (new additional links, edges) of a worse parent from three other vertices selected randomly [14].

3.3 Visualization Methods

Experimental data can be visualized in a few different ways, so a few typical visualizations are depicted here. The principles of the algorithm used are based on the philosophy of competition or offspring creation. Typical visualizations can appear as shown in Figures 1 and 2. At first glance, the interactions between individuals create structures that look like CNs. However, we have encountered results with visualizations that look like CNs, but after calculating network characteristics we found that some of those networks did not belong to the class of CNs with small-world phenomena. The vertices in Figures 1 and 2 are colored by the ratio of incoming and outgoing edges: a small vertex (small gray [pink] with dashed edges) has fewer incoming edges than outgoing, white (middle-sized) vertex is balanced (i.e., has the same number of incoming 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 with the maximum number of incoming edges. In EA jargon, the small vertex is an individual that has been used more for offspring creation than as a successful parent, and pink vertices reflect the opposite.

To ensure that an algorithm and the dynamics being investigated can really be understood and modeled as a CN, typical characteristics have been calculated and visualized such as the distributions of vertices degree in Figures 3 and 4.
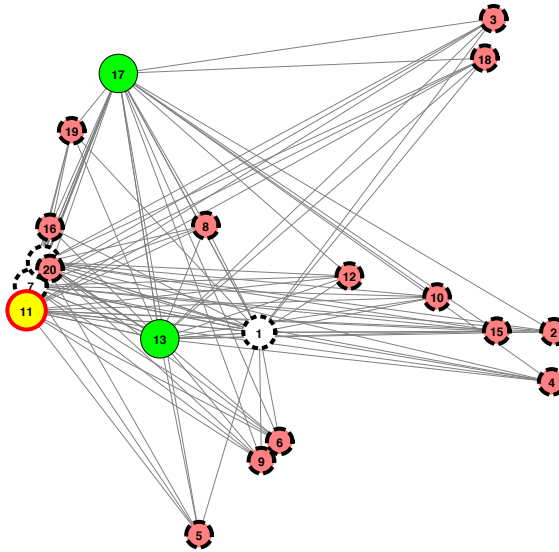
Figure 1. CN example of SOMA dynamics in a natural format. Vertex-individual 10 is the most profitable vertex. Display of multiple edges is disabled in the figures for better visualization.
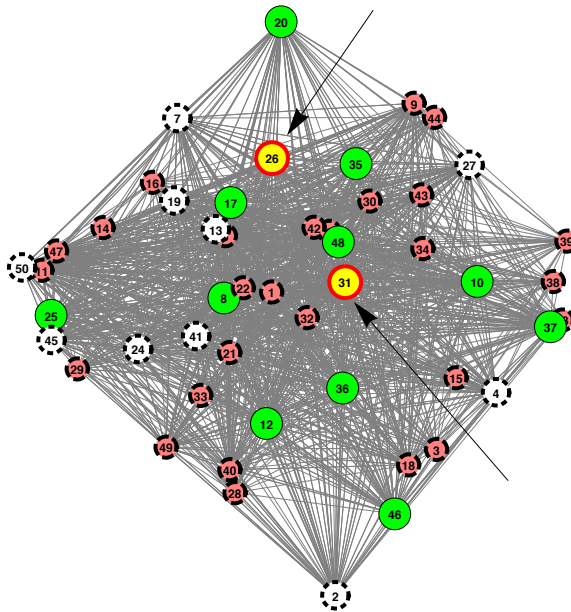

Figure 2. Visualization of the EA dynamics in the form of a CN. Each node represents an individual from the population, each edge (multiple edges are not depicted) depicts interaction with other individuals in the population.
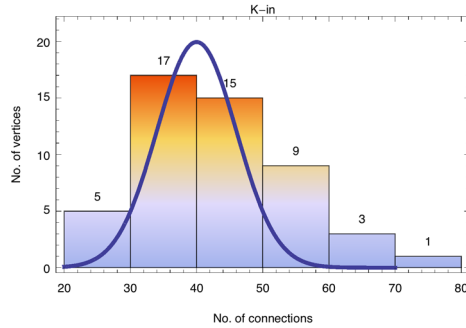
Figure 3. An example of DERand1Bin exhibiting normal-like distribution of vertices degree. No CNs were observed behind the evolutionary dynamics.
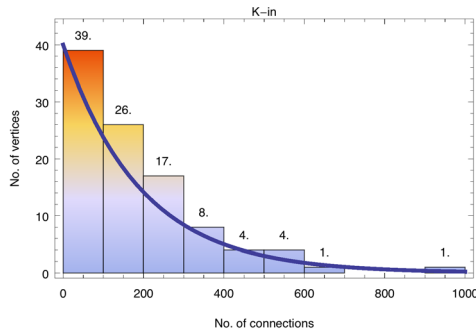
Figure 4. An example of a histogram that exhibits a long-tail distribution of vertices degree. This is a typical result for SOMA swarm-like algorithms.

4. Results

As reported earlier, both algorithms, in 10 versions, were tested on various functions to reveal their complex dynamics with a constant level of test function dimensionality (i.e., individual length) and different number of generations (migrations). All data was processed graphically and is shown alongside calculations of basic statistical properties as in Figures 3 through 8. The emergence of CN structure behind evolutionary dynamics depends on many factors. However, some special versions of the algorithms did not show CN structure despite the fact that the number of generations was quite large. Figures 3, 4, and 8 show histograms of vertex connections. Figure 7 depicts the history of the best individuals-vertices during the evolutionary process. Figures 5 and 6 show the history of 100 repeated simulations of CN structure formation (compare with Figure 7, where only one simulation is recorded). Some ideas about the results are discussed in Section 5.
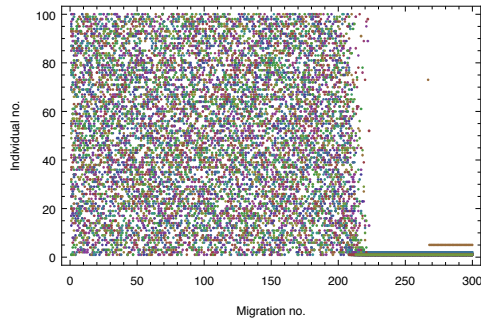
Figure 5. One hundred repeated simulations of CN structures forming on equation (2).
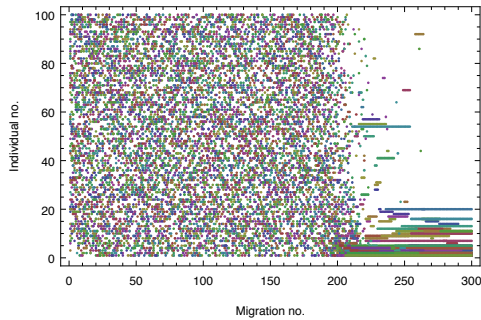

Figure 6. One hundred repeated simulations of CN structures forming on equation (1).
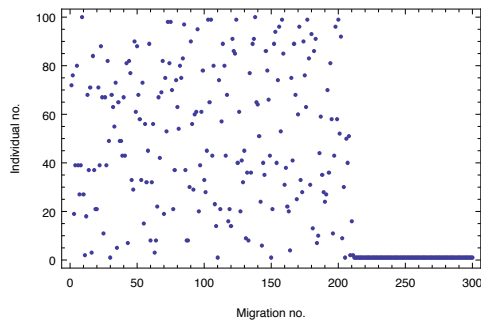

Figure 7. Activated leaders (*y* axis) with dependence on migrations (*x* axis) SOMA. Note that with migration between 200 and 300, the winner is still the first individual in the population.
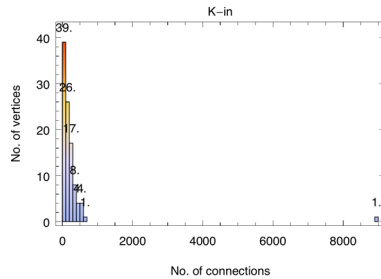
Figure 8. Histogram of the vertex connections based on a different experiment. Thanks to the fact that the first individual is still selected as the winner, it gets a lot of connections (8700) and the histogram does not look like a typical long-tail distribution.

5. Conclusion

The main motivation of this research is to determine whether it is possible to visualize and simulate underlying dynamics of an evolutionary process as a complex network (CN). Based on the preliminary results of only two algorithms with 10 versions and 16 out of 17 test functions, the following can be stated.

1. *Number of generations*. The occurrence of CN structure depends on the number of generations. With a small number of generations, no CN structure was established. This effect can be easily understood because a small number of generations means that evolutionary algorithms (EAs) do not have enough time to establish CN structure. This observation is quite logical when CN structure is not observable at the beginning of the linking process. During our experiments, the moment of establishing a CN structure was observed to depend on cost function dimension, population size, the algorithm used, and cost function. Very generally, EAs searching for a global extreme seem quite random at the beginning and when the global extreme domain is discovered, then a CN structure is quite quickly established.

2. *Dimensionality*. The impact on CN structure forming has been observed when the dimension of the cost function was big and the number of generations was too low. The selected EA was not able to finish the global extreme search successfully so all connections were not 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 the "rich becoming richer," that is, one vertex (individual) wins repeatedly (see Figures 7 and 8). This moment usually means that the global extreme has been found and further searching is not necessary.

3. *Test functions*. The dependence of the test function on CN structure forming was not strictly observed. The general consensus is that in more complex test functions, such as Schwefel's (equation (1)), the algo-

rithm needs more generations to establish a CN structure, that is, more complex functions require more generations and/or a bigger population size. In the case of simpler functions such as the first De Jong and low dimensions, the global extreme is quickly found and the phase of CN structure creation is very short, which activates the last phase of the rich becoming richer (see Figures 5 through 8).

4. *Population size*. CN structure formation was usually observed for populations of 100 or more individuals for 50 dimensions. Again, it is a parameter that alone does not influence CN structure formation, but does in combination with other parameters.

5. *Algorithm used*. CN structure formation 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 CN structure formation (in principle, each individual is selected to be a parent); see Figure 3. In the case of DELocalToBest, in which the best solution in the population plays an important role, CN structures have been observed, as well as in the SOMA strategies; Figure 4. The conclusion reached is that CN structure formation is more likely observed with swarm-like algorithms rather than "randomly remoted" algorithms. We think that this is quite logical and close to the idea of preferred linking in CNs that model social behavior such as citation networks.

The possible use and benefit of this approach, according to us, is as a novel approach to evaluate and control evolutionary dynamics. Based on the numerically demonstrated fact (no mathematical proof has been made) that EA dynamics can be visualized like CNs, we believe that there is a new area of research for the study of EA dynamics and their control via CN techniques of control [16]. Another domain of research is to study information flow in such a network by means of so-called *k*-shell decomposition.

A further issue of this research is whether EA dynamics, visualized as CNs, can be used elsewhere. Our answer is yes. Based on the philosophy of so-called coupled map lattice (CML) systems described in [17], it is possible to visualize CNs like a CML system (let us call it here CNSCML). In fact, we are able to visualize EA dynamics in this way like a CML system. Figure 9 can be used as an example where a zoom of a CNSCML is depicted. Each row represents one individual (in EA) or vertex (in CN visualization). Different colors represent different levels of vertex (individual) activations. As numerically demonstrated and proven in [17–20], it is possible to control such a system. This means that there is a real possibility of controlling the dynamics of EAs (CNs) via CML control philosophy.

For now, this paper has proposed a method for visualizing EA dynamics as a CNSCML. Our next research will be focused on the following.

1. Collection of the artificially generated data capturing CN dynamics as well as data from existing real CNs.

2. Use of results from our previous research, in which we have shown that EA dynamics can be modeled as a CN (EACN). We would like to convert EACN to the EACNCML and try to control the observed EA dynamics in order to increase performance.

3. Based on data from items 1 and 2, an exact analysis of when chaotic regimes are observed in CNCML will be done and also an analysis of what kind of routes to chaos exist (intermittence, crisis, bifurcations).

4. Use of selected EAs on CNSCML control with reverse investigation; what exact control has such an impact on the original CN structure (control here basically means that EAs have to change the structure of a CN).
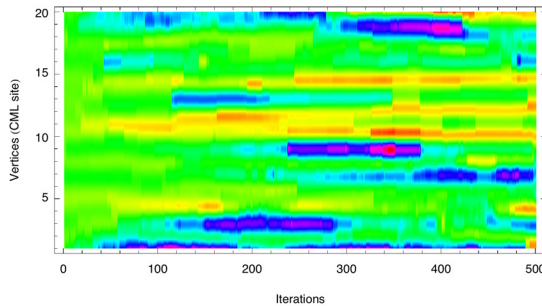

Figure 9. A zoom of EA dynamics as a CML system (i.e., EA → CN → CML) with 20 vertices in 500 iterations.

References

[1] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of Networks," *Advances in Physics*, **51**, 2002 pp. 1079–1187. arXiv:cond-mat/0106144.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex Networks: Structure and Dynamics," *Physics Reports*, **424**(4–5), 2006 pp. 175–308. doi:10.1016/j.physrep.2005.10.009.

[3] M. S. Granovetter, "The Strength of Weak Ties," *American Journal of Psychology*, **78**(6), 1973 pp. 1360–1380.

[4] L. C. Freeman, "Centrality in Social Networks: Conceptual Clarification," *Social Networks*, **1**, 1979 pp. 215–239.

[5] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge: Cambridge University Press, 1994.

[6] J.-P. Eckmann, E. Moses, and D. Sergi, "Entropy of Dialogues Creates Coherent Structures in E-mail Traffic," *Proceedings of the National Academy of Sciences of the United States of America*, **101**(40), 2004 pp. 14333–14337. doi:10.1073/pnas.0405728101 .

[7] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, M. Argollo de Menezes, K. Kaski, A.-L. Barabási, and J. Kertész, "Analysis of a Large-Scale Weighted Network of One-to-One Human Communication," *New Journal of Physics*, **9**(7), 2007 pp. 179. doi:10.1088/1367-2630/9/6/179.

[8] S. H. Lee, P.-J. Kim, Y.-Y. Ahn, and H. Jeong. "Googling Social Interactions: Web Search Engine Based Social Network Construction." (Oct 17, 2007) arXiv:0710.3268v1.

[9] A. Turing, "Intelligent Machinery," 1948, unpublished report for National Physical Laboratory. Published in *The Collected Works of A. M. Turing, Vol. 3* (P. T. Saunders, ed.), Amsterdam: North-Holland, 1992.

[10] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Ann Arbor: University of Michigan Press, 1975.

[11] H. Schwefel, "Numerische Optimierung von Computer-Modellen," Ph.D. thesis, 1974. Published in *Zeitschrift für Angewandte Mathematik und Mechanik*, **60**(5), 1980 p. 272. doi:10.1002/zamm.19800600516.

[12] I. Rechenberg, "Evolutionsstrategie-Optimierung technischer Systeme nach Prinzipien der biologischen Evolution," Ph.D. thesis, 1971. Published by Fromman-Holzboog, 1973.

[13] D. B. Fogel, "Unearthing a Fossil from the History of Evolutionary Computation," *Fundamenta Informaticae*, **35**(1–4), 1998 pp. 1–16.

[14] K. Price, "An Introduction to Differential Evolution," in *New Ideas in Optimization* (D. Corne, M. Dorigo, and F. Glover, eds.), London: McGraw-Hill, 1999 pp. 79–108.

[15] I. Zelinka, "SOMA—Self-Organizing Migrating Algorithm," in *New Optimization Techniques in Engineering* (G. C. Onwubolu and B. V. Babu, eds.), New York: Springer-Verlag, 2004.

[16] S. Meyn, *Control Techniques for Complex Networks*, Cambridge: Cambridge University Press, 2007.

[17] C. Grebogi and Y. C. Lai, "Controlling Chaos," in *Handbook of Chaos Control* (H. G. Schuster, ed.), New York: Wiley-VCH, 1999.

[18] I. Zelinka, "Real-Time Deterministic Chaos Control by Means of Selected Evolutionary Algorithms," *Engineering Applications of Artificial Intelligence*, **22**(2), 2008. doi:10.1016/j.engappai.2008.07.008.

[19] I. Zelinka, "Investigation on Realtime Deterministic Chaos Control by Means of Evolutionary Algorithms," in *1st IFAC Conference on Analysis and Control of Chaotic Systems (2006)*, Reims, France (W. Kang, M. Djemai, and N. Manamanni, eds.), 2006.

[20] R. Senkerik, I. Zelinka, and E. Navratil, "Optimization of Feedback Control of Chaos by Evolutionary Algorithms," in *1st IFAC Conference on Analysis and Control of Chaotic Systems (2006)*, Reims, France (W. Kang, M. Djemai, and N. Manamanni, eds.), 2006.