

# Network Visualization of Population Dynamics in the Differential Evolution

Petr Gajdoš, Pavel Krömer, Ivan Zelinka

Department of Computer Science  
VŠB Technical University of Ostrava  
Ostrava, Czech Republic

Email: {petr.gajdos, pavel.kromer, ivan.zelinka}@vsb.cz

**Abstract**—The dynamics of populational metaheuristic algorithms, such as the differential evolution, can be represented by evolving complex networks. The differential evolution is a widely-used real parameter optimization method with excellent results and many real-world applications. The search for hidden relationships, behaviors, and patterns in complex networks representing populational metaheuristics can provide an interesting information about the underlying optimization processes. Various methods for visual network investigation and mining became very popular in the last decade and represent a natural set of tools for such analyses. Here, we introduce a new approach for the visual analysis of such network with a special emphasis on network readability. The proposed method is universal and can be applied to any type of complex network modelling any algorithm applied to any problem.

## I. INTRODUCTION

The interpretation of the dynamic aspects of evolutionary algorithms as a complex network of interactions between candidate solutions (population members) and the analysis of static and dynamic properties of such network is a recent research trend [19]. The investigation of the relations between attributes of such networks and the properties of the modelled algorithms opens a number of research questions and provides a new class of instruments for a more efficient control of nature-inspired metaheuristic methods [19]. Initial studies were already presented for several variants of the differential evolution algorithm applied e.g. to the flowshop scheduling problem [2] and the permutative flowshop scheduling problem [3].

Visualization is an important and powerful step in the analysis of networks and graph structures. Visual methods represent complex networks in a graphical form suitable for machine and human analysis. In this work, we model the differential evolution applied to a collection of high-dimensional benchmarking functions by a complex evolving network and employ the Sammon's mapping and Edge bundling methods for its intuitive visual representation. A set of computational experiments is conducted to demonstrate the properties and quality of the proposed approach.

The remainder of this article is structured as follows: related work on network visualization is briefly summarized in sec-

tion II. The differential evolution is outlined in section III. The algorithms of Sammon's mapping and Edge bundling are described in section IV. The proposed visualization procedure is detailed in section V which also includes a number of illustrative examples of visualization results. Finally, the work is concluded and future research directions are outlined in section VI.

## II. COMPLEX NETWORK DYNAMICS

The majority of biological and natural systems such as ant colonies, various types of swarms, and neural networks, as well as human-made artificial networks and linked structures, can be interpreted as complex networks [1], [8]. Network models are routinely used for the analysis of various data-intensive domains such as social networks, World Wide Web, wireless sensor networks, mobile ad-hoc networks, protein-protein interaction and so on. However, they can be also employed to describe and study bio-inspired computational models [19].

Complex dynamic networks can be described by graphs that consist of a huge number of dynamic units represented by nodes and relationships represented by edges. Current research efforts in the area of complex dynamic networks are focused on the observation of behavioural patterns and on the development of accurate models that can mimic or simulate dynamic processes of the network. Several visualization techniques can be used to handle and present a complex graph illustrating population dynamics of a metaheuristic populational algorithm such as the differential evolution [14], [17]. Most of them can visualize the underlying temporal processes in the form of image sequences or movies. A static approach based on Sammon's mapping [15], [13] and edge bundling visualization [10], [9], [7] is proposed in this article. The proposed method mixes the similarity of network units with the graph topology. The method is demonstrated on the differential evolution algorithm, but it is universal and can be used to illustrate any type of complex network modelling any algorithm.

## III. DIFFERENTIAL EVOLUTION

The differential evolution (DE) is a versatile and easy to use stochastic evolutionary optimization algorithm [14]. It is

a population-based optimizer that evolves a population of real encoded vectors representing the solutions to given problem. The DE was introduced by Storn and Price in 1995 [16] and it quickly became a popular alternative to the more traditional types of evolutionary algorithms. It evolves a population of candidate solutions by iterative modification of candidate solutions by the application of the differential mutation and crossover [14]. In each iteration, so called trial vectors are created from current population by the differential mutation and further modified by various types of crossover operator. At the end, the trial vectors compete with existing candidate solutions for survival in the population.

#### A. Traditional differential evolution

The DE starts with an initial population of  $N$  real-valued vectors. The vectors are initialized with real values either randomly or so, that they are evenly spread over the problem space. The latter initialization leads to better results of the optimization [14]. During the optimization, the DE generates new vectors that are scaled perturbations of existing population vectors. The algorithm perturbs selected base vectors with the scaled difference of two (or more) other population vectors in order to produce the trial vectors. The trial vectors compete with members of the current population with the same index called the target vectors. If a trial vector represents a better solution than the corresponding target vector, it takes its place in the population [14].

The two most significant parameters of the DE are scaling factor and mutation probability [14]. The scaling factor  $F \in [0, \infty]$  controls the rate at which the population evolves and the crossover probability  $C \in [0, 1]$  determines the ratio of components that are transferred to the trial vector from its opponent. The size of the population and the choice of operators are other important parameters of the optimization process.

The basic operations of the classic DE can be summarized using the following formulas [14]: the random initialization of the  $i$ th vector with  $N$  parameters is defined by

$$x_j^i = \text{rand}(b_j^L, b_j^U), \quad j \in \{1, \dots, N\} \quad (1)$$

where  $b_j^L$  is the lower bound of  $j$ th parameter,  $b_j^U$  is the upper bound of  $j$ th parameter and  $\text{rand}(a, b)$  is a function generating a random number from the range  $[a, b]$ . A simple form of standard differential mutation is given by

$$\mathbf{v}^i = \mathbf{v}^{r1} + F(\mathbf{v}^{r2} - \mathbf{v}^{r3}) \quad (2)$$

where  $F$  is the scaling factor and  $\mathbf{v}^{r1}$ ,  $\mathbf{v}^{r2}$  and  $\mathbf{v}^{r3}$  are three random vectors from the population. The vector  $\mathbf{v}^{r1}$  is the base vector,  $\mathbf{v}^{r2}$  and  $\mathbf{v}^{r3}$  are the difference vectors, and the  $i$ th vector in the population,  $\mathbf{v}^i$  is the trial vector. It is required that  $i \neq r1 \neq r2 \neq r3$ .

An alternative differential mutation which favours exploitation over exploration is defined by

$$\mathbf{v}^i = \mathbf{x}^{\text{best}} + F(\mathbf{v}^{r1} - \mathbf{v}^{r2}) \quad (3)$$

and combines two randomly chosen difference vectors with the best vector in population,  $\mathbf{x}^{\text{best}}$ .

The uniform (binomial) crossover that combines the target vector,  $\mathbf{x}^i$ , with the trial vector,  $\mathbf{v}^i$ , is given by

$$v_j^i = \begin{cases} v_j^i & \text{if } (\text{rand}(0, 1) < C) \text{ or } j = j_{\text{rand}} \\ x_j^i & \text{otherwise} \end{cases} \quad (4)$$

for each  $j \in \{1, \dots, N\}$ . The random index  $j_{\text{rand}}$  is in the above selected randomly as  $j_{\text{rand}} = \text{rand}(1, N)$ . The uniform crossover replaces the parameters in  $\mathbf{v}^i$  by the parameters from the target vector  $\mathbf{x}^i$  with probability  $1 - C$ . The outline of the classic DE according to [6], [14] is summarized in Algorithm 1.

---

#### Algorithm 1: A summary of classic Differential Evolution

---

```

1 Initialize the population  $P$  consisting of  $M$  vectors using eq. (1);
2 Evaluate an objective function ranking the vectors in the population;
3 while Termination criteria not satisfied do
4   Let  $G =$  number of current generation;
5   for  $i \in \{1, \dots, M\}$  do
6     Differential mutation: Create trial vector  $\mathbf{v}^i$  according
       to eq. (2);
7     Validate the range of coordinates of  $\mathbf{v}^i$ . Optionally adjust
       coordinates of  $\mathbf{v}^i$  so, that it is a valid solution to a given
       problem;
8     Perform uniform crossover. Select randomly one parameter
        $j_{\text{rand}}$  in  $\mathbf{v}^i$  and modify the trial vector using eq. (4);
9     Evaluate the trial vector.;
10    if trial vector  $\mathbf{v}^i$  represent a better solution than target vector
        $\mathbf{x}^i$  then
11      | add  $\mathbf{v}^i$  to  $P^{G+1}$ 
12    else
13      | add  $\mathbf{x}^i$  to  $P^{G+1}$ 
14    end
15  end
16 end

```

---

## IV. VISUALIZATION METHODS

Two advanced visualization methods were in this work employed to obtain well-understandable, accurate, and visually appealing graphical models of complex network representations of the DE algorithm.

#### A. Sammon's projection

The Sammon's projection [15] is one of several existing methods for projecting a data set from an original, high-dimensional data space to space with lower dimensionality. It aims at preserving the between-point distances from the high-dimensional data space in the lower-dimensional projection space. This goal is achieved by minimizing an error criteria that penalizes the changes of a distance between points in the original high-dimensional data space and in the low-dimensional projection space. For the purpose of visual data analysis, projections into two and three-dimensional spaces (2D and 3D) are of utmost importance.

Suppose that we have a collection  $X$  with  $p$  data points  $X = \{X_1, X_2, \dots, X_p\}$  where each data point  $X_i$  is an  $m$ -dimensional vector  $X_i = \langle x_{i1}, x_{i2}, \dots, x_{im} \rangle$ . At the same time

we define a collection  $Y$  of  $p$  data points  $Y = \{Y_1, Y_2, \dots, Y_p\}$  where each data point  $Y_i$  is a  $k$ -dimensional vector and  $k < m$ . The initial values of the coordinates in  $Y_i$  are chosen at random. The distance between vectors  $X_i$  and  $X_j$  is denoted  $d_{ij}^*$  while the distance between corresponding vectors  $Y_i$  and  $Y_j$  in the lower-dimensional space is denoted  $d_{ij}$ . Here, any distance measure can be used to evaluate  $d_{ij}$ . However, the distance measure suggested originally by Sammon is the traditional Euclidean metric [15]. In that case,  $d_{ij}^*$  and  $d_{ij}$  are defined in the following way:

$$d_{ij}^* = \sqrt{\sum_{k=1}^m (x_{ik} - x_{jk})^2}, \quad (5)$$

$$d_{ij} = \sqrt{\sum_{k=1}^d (y_{ik} - y_{jk})^2}. \quad (6)$$

Projection error  $E$  (so-called Sammon's stress) measures how well the current configuration of  $p$  data points in the  $k$ -dimensional space matches the  $p$  points in the original  $m$ -dimensional space.

$$E = \frac{1}{\sum_{i < j} (d_{ij}^*)} \sum_{i < j} \frac{(d_{ij}^* - d_{ij})^2}{d_{ij}^*} \quad (7)$$

In order to minimize the projection error,  $E$ , any minimization technique can be used. Sammon's original paper from 1969 [15] used widely known methods such as pseudo-Newton (steepest descent) minimization:

$$y'_{ik}(t+1) = y'_{ik}(t) - \alpha \frac{\frac{\partial E(t)}{\partial y'_{ik}(t)}}{\left| \frac{\partial^2 E(t)}{\partial y'_{ik}(t)^2} \right|}, \quad (8)$$

where  $y'_{ik}$  is the  $k^{\text{th}}$  coordinate of the data point's position  $y'_i$  in the projected low-dimensional space. The constant  $\alpha$  is usually taken from a range 0.3 – 0.4, originally proposed by Sammon. However, this range is not optimal for all types of problems. Equation (8) can cause a problem at the inflection points where the second derivative is very small. Therefore, the gradient descent may be used as an alternative minimization method. An illustrative *CUBE* example of the Sammon's projection shows a set of points laying on six cube faces (see Figure 1) projected into 2D screen space (see Figure 2).

### B. Edge bundling visualization

A suitable visualization technique must be chosen to be able to visualize the dynamics of a complex network. The usual node-link graph layouts represent a family of intuitive and well-understandable graph drawing procedures. However, the readability of such layouts is highly affected by the number of nodes and edges in the graph. There exist several advanced layouting algorithms [4], [20], but the mentioned common problem still arises e.g. in case of dense graphs. If there is no way to reduce the number of graph elements and graph layouting does not improve the readability, some

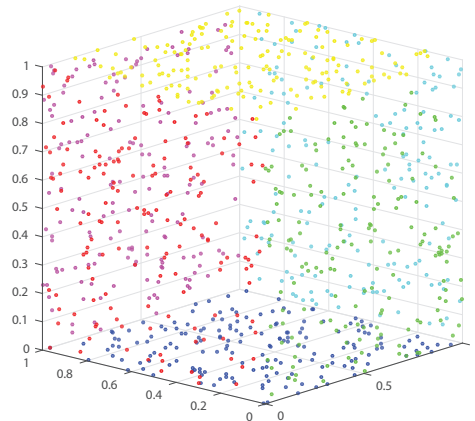


Fig. 1. A 3D view of the *CUBE* example data set before Sammon's projection.

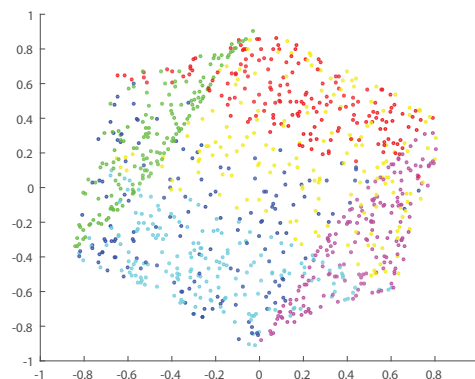


Fig. 2. A 2D view of the *CUBE* example data set after Sammon's projection.

visualization techniques can help to obtain better outputs. The edge bundling technique and its extensions were introduced in [7] and used to visualize various data sets in different application domains [7], [9], [12]. A hierarchical edge bundling method based on Sammon's projection was used in the proposed method to achieve an accurate, readable, and visually appealing representation of the complex network representing the dynamics of the DE algorithm. It puts a special emphasis on the relationships between the tuple of nodes representing the three parent nodes and the child node that take part in differential mutation and crossover in the  $/DE/rand/1$  differential evolution. The graph  $G$  can be defined as follows:

- Let  $G = (N, E)$  be a graph that consists of a set of nodes  $N$  and a set of edges  $E$ .
- An edge  $e \in E$  is an ordered pair of elements of  $N$ , where  $E \subset (N \times N)$ .
- For all edges  $e = [n_0, n_1]$ , where  $n_0 \in N$  and  $n_1 \in N$ , let  $n_0$  is called a *child* node and  $n_1$  is called a *parent* node. Let  $E \subset (N \times N)$  and  $e \in E$  and  $n_0 \neq n_1$ .
- A node  $n \in N$  represents an  $m$ -dimensional data vector

projected into a 2D space by Sammon’s projection, such that  $n = [\mathbb{R}, \mathbb{R}]$ . Then  $n = [x, y]$ , where  $x \in \mathbb{R}$  and  $y \in \mathbb{R}$  represent the screen space coordinates of the node  $n$ .

The primary goal of the use of edge bundling visualization consists in the improvement of graph readability. Figure 3 and Figure 4 illustrate the same graph  $G$  with and without edge bundling, respectively. Usually, the edges are represented by some kind of splines that can be modified in terms of number of control points, edge width, etc. The “bundling” process runs for a defined number of iterations. During them, a physically based simulation splits the edges, creates control points, and moves spline control points close each other with respect to an edge similarity matrix. We refer to [9], [11] for more information on edge bundling techniques as this is out of scope of this article.

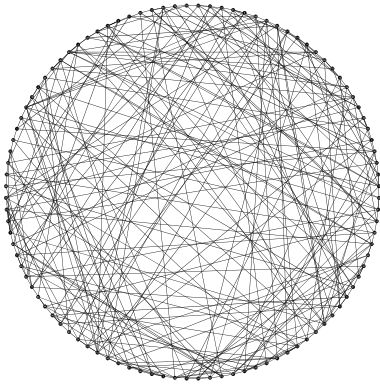


Fig. 3. An illustrative graph,  $G$ , displayed in a circular graph layout with direct edges.

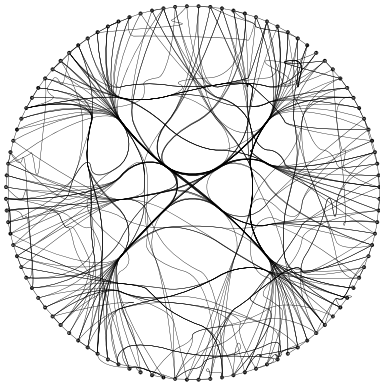


Fig. 4. An illustrative graph,  $G$ , displayed in a circular graph layout with edge bundling.

## V. VISUALIZATION PROCESS

The proposed visualization methods are demonstrated on a complex network model of a DE solving a test problem from a well-known collection of benchmarking functions [18].

### A. Test problem

For the sake of simplicity, all proposed visualization techniques are demonstrated on the “Ackley” function in ten dimensional space, i.e.  $m = 10$ .

$$f_1(\mathbf{x}) = -20 \cdot \exp \left( -0.2 \sqrt{\frac{1}{m} \cdot \sum_{i=1}^m x_i^2} \right) - \exp \left( \frac{1}{m} \sum_{i=1}^m \cos(2\pi x_i) \right) + 20 + e \quad (9)$$

where  $m$  is the problem dimension,  $\mathbf{x} = (x_1, \dots, x_m)$  is parameter vector,  $e \approx 2.71828$  is Euler’s number, and  $\exp(a) = e^a$  is exponential function.

It is one of the best-known and most used real-parameter optimization test functions in the area of bio-inspired computing [18], [5]. Its shape in 3D space is illustrated in Figure 5. As it can be seen, there are many *local saddle points (local optima)*. This function represents for all metaheuristic and bio-inspired methods a challenging test problem because a poor algorithm with a bad ability to escape local minima can easily miss the global optimum and remain entrapped in an attraction basin of one of its many local minima.

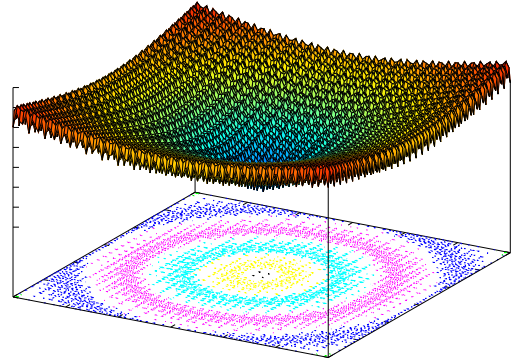


Fig. 5. The behavior of Ackley function in 3D.

### B. Proposed visualization procedure

Algorithm 1 summarizes the standard steps of a traditional DE. For the visualization, all DE iterations can be computed in advance before the visualization. Then, the process of visualization involves the following steps:

- *StartIdx* is chosen as an index of the first visualized iteration of DE.
- *IterCount* is defined as a number of visualized iterations.
- All individuals from the *StartIdx*-th up to *StartIdx* + *IterCount*-th iteration make an input for Sammon’s projection.
- Sammon’s projection is performed with defined number of iterations, and final 2D vectors are stored.

- A graph is constructed such that 2D vectors represent nodes in the 2D space and edges are reconstructed from the DE run. An edge is placed, or edge weight increased, in every generation between every pair of nodes that take part in the creation of a new solution (trial vector) that is better (i.e. has lower fitness) than the target vector.
- Edge bundling visualization technique is performed to achieve a more readable and visually appealing output in case of dense graphs.

### C. Evolution paths

As it was already mentioned, the proposed method is designed to visualize the behavior of an evolutionary algorithm. If the edge bundling technique is used, Sammon's projection can be employed to uncover and illustrate the improvements of individuals in time. In this case, all generations must be visualized together to be able to draw the paths of evolution (see Figure 6). The dynamics of the DE algorithm is expressed as a system of crossing evolution paths in the Sammon's projection space. A higher dynamics of the algorithm (i.e. a large number of successful interactions between population members) leads to a graph representation with a larger number of crossing paths and longer average distances between the nodes. Figure 6 A) represents an almost stable DE where the changes of fitness values are rare and small. Contrariwise, Figure 6 B) represents a DE in its early state where the individuals in the population (data vectors in  $m$ -dimensional space) change frequently and the improvements in their fitness values are significant.

Here, the graph  $G$  is defined as follows:

- Let  $G = (N, E)$  is a graph that consists of a set of nodes  $N$  and a set of edges  $E$ .
- $N = N_0 \cup N_1 \cup \dots \cup N_{g-1}$ , where  $g$  is the number of generations, and if  $i \in \langle 0, g-1 \rangle$  is a zero-based generation index then  $N_i$  is a set of individuals in the  $i$ -th generation.
- An edge  $e \in E$  is an ordered pair of elements of  $N$ , where  $E \subset (N_0 \times N_1) \cup (N_1 \times N_2) \cup \dots \cup (N_{g-2} \times N_{g-1})$ , where  $g$  is the number of generations.
- For all edges  $e = [n_0, n_1]$ , where  $n_0 \in N_{i-1}$  and  $n_1 \in N_i$  where  $i \in \langle 0, g-1 \rangle$ , let  $e$  is called an *path edge*, iff  $n_0$  represents the node in the  $(i-1)$ -th generation and  $n_1$  represents the SAME node in the  $i$ -th generation.
- A node  $n \in N$  represents an  $m$ -dimensional data vector projected into a 2D space by Sammon's projection, such that  $n = [\mathbb{R}, \mathbb{R}]$ . Then  $n = [x, y]$ , where  $x \in \mathbb{R}$  and  $y \in \mathbb{R}$  represent the screen space coordinates of the node  $n$ .

### D. Evolution graph

Now an *evolution graph* can be constructed. This graph illustrates the mutual influences in terms of evaluation of the population. Again,  $g$  generations must be visualized together to keep the information on similarity between individuals within and across all generations (see Figure 7). Sammon's projection was in this case applied to obtain clusters of nodes. Here, the

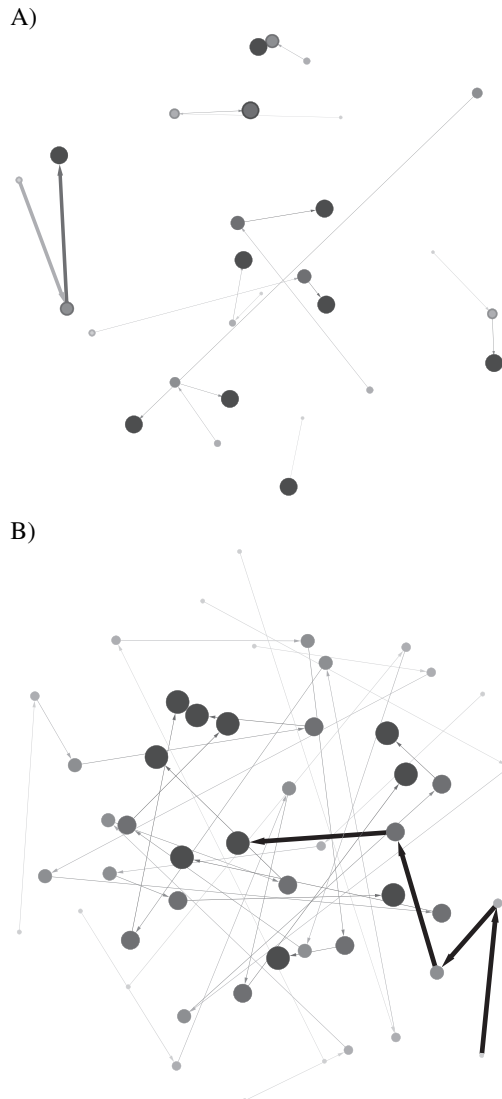


Fig. 6. An illustrative visualizations of the evolution paths of two individuals for five DE generations. The smaller size of the node, the older generation it corresponds to. The dynamics of the DE algorithm is expressed as a system of crossing evolution paths in the Sammon's projection space.

usage of Edge bundling technique is demonstrated in Figure 7 B), where the non-selected edges were physically bundled together to achieve better readability. Finally, Figure 8 illustrates a pure Edge bundling-based visualization with omitted Sammon's projection. Here, a well known force directed layout was used at the cost of losing valuable information about clusters of nodes.

In this case, the graph  $G$  is defined as follows:

- Let  $G = (N, E)$  be a graph that consists of a set of nodes  $N$  and a set of edges  $E$ .
- $N = N_0 \cup N_1 \cup \dots \cup N_{g-1}$ , where  $g$  is the number of generations, and if  $i \in \langle 0, g-1 \rangle$  is a zero-based generation index then  $N_i$  is a set of individuals in the

$i$ -th generation.

- An edge  $e \in E$  is an ordered pair of elements of  $N$ , where  $E \subset (N_0 \times N_1) \cup (N_1 \times N_2) \cup \dots \cup (N_{g-2} \times N_{g-1})$ , where  $g$  is the number of generations.
- For all edges  $e = [n_0, n_1]$ , where  $n_0 \in N_{i-1}$  and  $n_1 \in N_i$  where  $i \in \langle 0, g-1 \rangle$ , let  $e$  is called an *path edge*, iff  $n_0$  represents the node in the  $(i-1)$ -th generation and  $n_1$  represents a DIFFERENT parent node in the  $i$ -th generation.
- A node  $n \in N$  represents an  $m$ -dimensional data vector projected into a 2D space by Sammon's projection, such that  $n = [\mathbb{R}, \mathbb{R}]$ . Then  $n = [x, y]$ , where  $x \in \mathbb{R}$  and  $y \in \mathbb{R}$  represent the screen space coordinates of the node  $n$ .

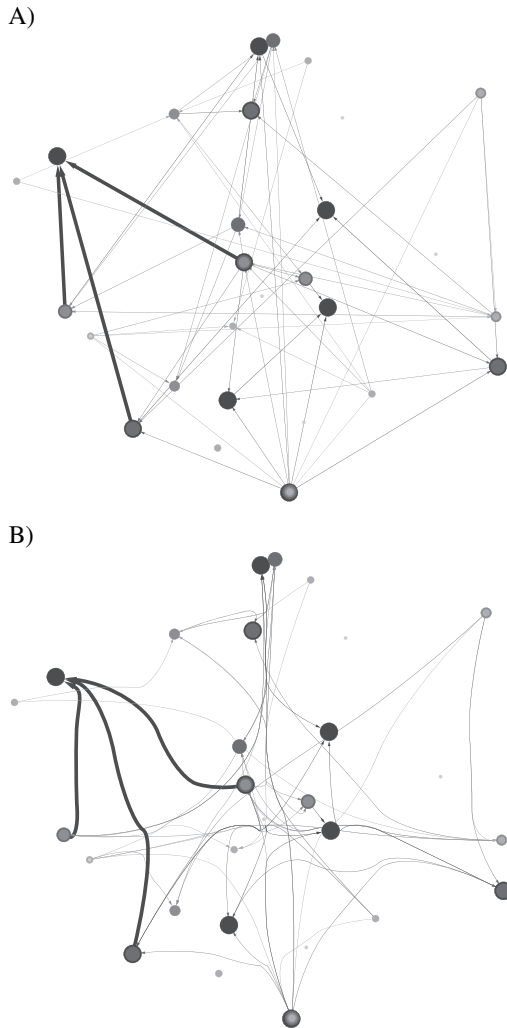


Fig. 7. An illustrative visualizations of the DE during five generations. All five generations were processed by the Sammon's projection. The smaller size of the node, the older generation of the node. Graph A) is without the edge bundling while graph B) demonstrates the visualization with edge bundling.

## VI. CONCLUSION AND FUTURE WORKS

This work proposes a novel method for visual rendering of a complex network representation of population dynamics

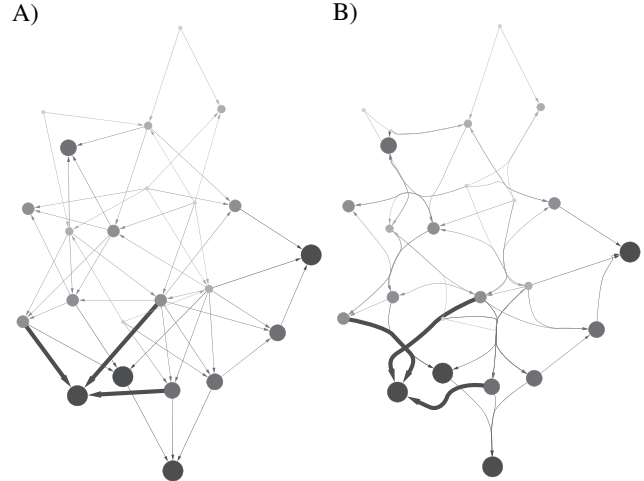


Fig. 8. An illustrative visualizations of the DE during five generations without Sammon's projection. A simple force-directed layout was used place nodes in 2D space. Graph A) is without the edge bundling while graph B) demonstrates the visualization with edge bundling.

in the differential evolution algorithm. The complex network representation allows casting the dynamics of a DE algorithm into a graph that can be studied and eventually used to control the algorithm. The proposed visual rendering, based on Sammon's projection and edge bundling, provides an intuitive, accurate, and visually appealing representation of the network.

The illustrative examples, provided in this work, demonstrate the proposed method and confirm that the properties of graph theory in combination with projection techniques can provide a valuable information regarding the status of the DE algorithm. The initial experiments show that the dynamics of evolutionary algorithm can be projected as a sequence of individual paths into 2D space with respect to similarity of data vectors. Further analysis of the generated graph can be used to control the DE e.g. by early detection of stagnation, prevention of local minima entrapment etc.

Last but not least, the physically based edge bundling technique can be used to increase graph readability and emphasize selected edges.

Network-based modeling and analysis is interesting also from a broader perspective since it is now well-understood that many of real-world processes can be mapped on network properties and complex network models may provide new insights into and accurate emulation of their behavior.

## ACKNOWLEDGEMENT

This work was supported by the Czech Science Foundation under the grant no. GACR P103/15/06700S and partially supported by grants of SGS, VŠB - Technical University of Ostrava, no. SP2015/146 and SP2015/142.

## REFERENCES

- [1] A.-L. Barabasi. *Linked - The new Science of Networks*. Perseus Publishing, 2002.

- [2] D. Davendra, I. Zelinka, M. Metlicka, R. Senkerik, and M. Pluhacek. Complex network analysis of differential evolution algorithm applied to flowshop with no-wait problem. In *Differential Evolution (SDE), 2014 IEEE Symposium on*, pages 1–8, Dec 2014.
- [3] D. Davendra, I. Zelinka, R. Senkerik, and M. Pluhacek. Complex network analysis of evolutionary algorithms applied to combinatorial optimisation problem. In P. Krömer, A. Abraham, and V. Snášel, editors, *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, volume 303 of *Advances in Intelligent Systems and Computing*, pages 141–150. Springer International Publishing, 2014.
- [4] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Surveys (CSUR)*, 34(3):313–356, 2002.
- [5] Q. Duan, V. K. Gupta, and S. Sorooshian. Shuffled complex evolution approach for effective and efficient global minimization. *Journal of optimization theory and applications*, 76(3):501–521, 1993.
- [6] A. Engelbrecht. *Computational Intelligence: An Introduction, 2nd Edition*. Wiley, New York, NY, USA, 2007.
- [7] O. Ersoy, C. Hurter, F. V. Paulovich, G. Cantareiro, and A. Telea. Skeleton-based edge bundling for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2364–2373, 2011.
- [8] D. M. Gordon. *Ant Encounters: Interaction Networks and Colony Behavior*. Primers in complex systems. Princeton University Press, 2010.
- [9] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.
- [10] D. Holten and J. J. Van Wijk. Force-directed edge bundling for graph visualization. In *Computer Graphics Forum*, volume 28, pages 983–990. Wiley Online Library, 2009.
- [11] U. Kang, L. Akoglu, and D. H. Chau. Big graph mining for the web and social media: algorithms, anomaly detection, and applications. In *WSDM*, pages 677–678, 2014.
- [12] R. Klapaukh, D. J. Pearce, and S. Marshall. Towards a vertex and edge label aware force directed layout algorithm. In *Proceedings of the Thirty-Seventh Australasian Computer Science Conference-Volume 147*, pages 29–37. Australian Computer Society, Inc., 2014.
- [13] J. Martinovic, P. Gajdos, and V. Snasel. Similarity in information retrieval. In *7th Computer Information Systems and Industrial Management Applications*, pages 145–150. IEEE, 2008.
- [14] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag, Berlin, Germany, 2005.
- [15] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, 18(5):401–409, May 1969.
- [16] R. Storn. Differential evolution design of an IIR-filter. In *Proceeding of the IEEE Conference on Evolutionary Computation ICEC*, pages 268–273. IEEE Press, 1996.
- [17] R. Storn and K. Price. Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical report, 1995.
- [18] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real Parameter Optimization. Technical report, Nanyang Technological University, 2005.
- [19] I. Zelinka, D. Davendra, J. Lampinen, R. Senkerik, and M. Pluhacek. Evolutionary algorithms dynamics and its hidden complex network structures. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 3246–3251, July 2014.
- [20] Y. Zhang. *Boundary Constraints in Force-Directed Graph Layout*. University of California, Santa Cruz, 2014.