A REVIEW ON THE SIMULATION OF SOCIAL NETWORKS INSIDE HEURISTIC ALGORITHMS

Roman Senkerik, Michal Pluhacek, Adam Viktorin, Tomas Kadavy, Jakub Janostik and Zuzana Kominkova Oplatkova

Tomas Bata University in Zlin, Faculty of Applied Informatics
Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{senkerik, oplatkova , pluhacek , aviktorin , kadavy}@fai.utb.cz

KEYWORDS
Social networks; Graphs; Analysis; Evolutionary algorithms; Swarm Algorithms.

ABSTRACT

This paper represents a comprehensive review of selected methods for visualization of the population dynamics of the swarm and evolutionary algorithms in the form of networks. The whole idea is based on the obvious similarity between interactions between individuals in a swarm and evolutionary algorithms and for example, users of social networks, society, etc. The analogy between individuals from the population used in an arbitrary evolutionary or swarm-based algorithm and vertices (nodes) of a network is discussed here, as well as between edges in a network and communication between individuals in a population. Simple experiments with four well-known heuristic algorithms are described here, giving an insight into different approaches to the building of the network during metaheuristic run.

INTRODUCTION

In this review paper, we have merged two different attractive areas of research: (complex) networks and evolutionary computation. Interactions in a swarm and evolutionary algorithms during the optimization process can be considered like user interactions in social networks or just people in society. It has been observed that networks generated by evolutionary dynamics show properties of complex networks in certain time frames and conditions (Skanderova et al. 2016).

Evolutionary computation is a sub-discipline of computer science belonging to the bio-inspired computing area. In recent decades, more robust and effective algorithms have been introduced. Like Differential Evolution (DE) (Das et al. 2016), Particle Swarm Optimization (PSO) (Engelbrecht 2010), Self Organizing Migrating Algorithm (SOMA) (Zelinka 2016), Artificial Bee Colony (Karaboga and Basturg 2007) or Firefly Algorithm (FA) (Fister et al. 2013).

Currently, the utilization of complex networks as a visualization tool for the analysis of population dynamics for evolutionary and swarm-based algorithms is becoming an interesting open research task. The population is visualized as an evolving complex network that exhibits non-trivial features – e.g., degree distribution, clustering, and centralities. These features offer a clear description of the population under evaluation and can be utilized for the adaptive population as well as parameter control during the metaheuristic run. The initial studies (Zelinka et al. 2014; Davendra et al. 2014) describing the possibilities of transforming population dynamics into complex networks were followed by the successful adaptation and control of the metaheuristic algorithm during the run through the given complex networks´ frameworks (Skanderova and Fabian 2015; Metlicka and Davendra 2015; Gajdos et al. 2015; Janostik et al. 2015).

This research paper reviews the complex network frameworks for DE, PSO, FA and Fireworks algorithm (FWA) (Tan and Zhu 2010). Currently, all algorithms above are known as powerful metaheuristic tools for solving optimization problems.

The organization of this paper is as follows: Firstly, the motivation, background of the heuristic algorithms and the concept of complex network framework for heuristics are briefly described; followed by the simple experiment designs, graphical visualizations, and conclusions.

MOTIVATION AND RELATED WORKS

This paper represents a comprehensive overview and continuation of the previous successful initial experiments, which are referred in particular sections. The motivation for the research presented herein can be summarized as follows:

- To show the different approaches in building complex networks to capture the dynamics either of evolutionary or swarm-based algorithms.
- To investigate the time development of the influence of either individual selection inside a DE or (density of) communication inside a swarm transferred into the complex network.
- To briefly discuss the possible utilization of complex network attributes that can be extracted from graph visualizations – e.g., adjacency graphs, centralities, clustering, etc for adaptive population and parameter control during the metaheuristic run.

HEURISTIC ALGORITHMS

This section contains the basic background for the metaheuristic algorithms DE, PSO, FA, and FWA that were used in this review paper. The algorithms workflows are limited to the minimum information required for the better understanding of social

Proceedings 32nd European Conference on Modelling and Simulation ©ECMS Lars Nolle, Alexandra Burger, Christoph Tholen, Jens Werner, Jens Wellhausen (Editors)
ISBN: 978-0-9932440-6-3/ ISBN: 978-0-9932440-7-0 (CD)

interactions capturing to the network. More details about the algorithms can be found in the referred original literature sources.

Differential Evolution

DE is a population-based optimization method that works on real-number-coded individuals (Das et al. 2016). DE is quite robust, fast, and effective, with global optimization ability. There are essentially five inputs to the heuristic. D is the size of the problem, G_{max} is the maximum number of generations, NP is the total number of solutions, F is the scaling factor of the solution and CR is the factor for crossover. F and CR together make the internal tuning parameters for the heuristic.
In this research, we have used original DE "rand/1/bin" (1) mutation strategy and binomial crossover. The parent indices (vectors) are selected by standard PRNG with uniform distribution. Mutation strategy "rand/1" uses three random parent vectors with indexes $r1$, $r2$, and $r3$, where $r1 \neq r2 \neq r3$. Mutated vector $v_{i,G}$ is obtained from three different vectors $\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$ from current generation G with the help of with the help of scaling factor F_i as follows (1):

$$\boldsymbol{v}_{i,G} = \boldsymbol{x}_{r1,G} + F_i\left(\boldsymbol{x}_{r2,G} - \boldsymbol{x}_{r3,G}\right) \tag{1}$$

After the mutation is done, the crossover procedure based on the defined and fixed CR value is performed between active individual from the population and newly created mutated vector. If both processes lead to the better solution, this one will be stored in the next generation G+1 replacing the active individual (solution).

PSO Algorithm

Original PSO algorithms take their inspiration from the behavior of fish and birds (Engelbrecht 2010). The knowledge of the global best-found solution (*gBest*) is shared among the particles in the swarm. Furthermore, each particle knows its own (personal) best-found solution (*pBest*). The last important part of the algorithm is the velocity of each particle, which is taken into account during the calculation of the particle´s movement. The new position of each particle is then given by (2), where $\boldsymbol{x}_i^{t+1}$ is the new particle position; $\boldsymbol{x}_i^t$ refers to the current particle position, and $\boldsymbol{v}_i^{t+1}$ is the new velocity of the particle. To calculate the new velocity, the distance from *pBest* and *gBest* is taken into account along with its current velocity (3), where c_1, c_2 represents the acceleration constants, and symbol j points to the j-th component of the dimension D..

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{2}$$

$$\begin{aligned} v_{ij}^{t+1} &= v_{ij}^t + c_1 \cdot Rand \cdot (pBest_{ij} - x_{ij}^t) + \\ &+ c_2 \cdot Rand \cdot (gBest_j - x_{ij}^t) \end{aligned} \tag{3}$$

Firefly Algorithm

FA was firstly introduced by X. S. Yang (Yang 2010). This nature-based algorithm tries to simulate the mating behavior of fireflies at night. Every firefly emits flashing light to lure appropriate mating partner. For the formulation of the FA, the flashing light is associated with the objective function value that is optimized. For simplicity, the three following rules are used:

- All fireflies are sexless (each firefly can attract, or be attracted by, any of the remaining ones).
- The attractiveness of fireflies is proportional to their brightness. Thus the less bright firefly will move toward the brighter one. The brightness decreases with the distance between fireflies. If there is a no brighter firefly, the particular one will move randomly.
- The firefly brightness is based on the objective function value.

The brightness of firefly consists of three factors: the objective function value, the distance between two compared fireflies and absorption of media in which the fireflies are.

Fireworks Algorithm

The FWA is an algorithm that is inspired by fireworks explosion in a night sky. This algorithm is initialized with a random population of fireworks. The particular firework position is represented as coordinates in n-dimensional space of solutions. These coordinates are parameters of the optimized problem. The number of the fireworks is defined by the parameter NP. This algorithm consists of four parts: explosion operator, mutation operator, mapping rule and selection strategy. These parts and adjustable parameters are more explained in next sections. The realization of FWA is as follows:

- Randomly generate NP fireworks in the n-dimensional search space.
- Obtain fitness values of these generated fireworks by the fitness function.
- Calculate the number of generated sparks and their amplitude for each firework by explosion operator.
- Use Gaussian mutation to generate new random sparks by mutation operator.
- Apply mapping rule to all generated sparks.
- Calculate fitness values of sparks, and by applying selection, strategy pick the selected sparks as new fireworks.
- If the terminal conditions are met, stop the algorithm. Otherwise, continue the iteration process from the third step.

COMPLEX (SOCIAL) NETWORKS

A complex network (CN) is a graph which has unique properties - usually in the real-world graph domain. A complex network contains features which are unique to the assigned problem. These features are important markers for a population used in Evolutionary/Swarm

based algorithms (Davendra et al. 2016). The following two described non-trivial features are important for a quick analysis of the network thus created.

Degree Centrality is defined as the number of edges connected to a specific node. Degree Centrality is an important distribution hub in the network since it connects - and thereby, distributes most of the information flowing through the network. Using Degree Centrality, we could analyze if stagnation or premature convergence is occurring within the population. From the graphs, it can be seen either that the multiple nodes are increasing (emphasizing their prominence in the population – helping in generating of the better individuals) or the degree centrality values are stagnating (i.e., no improvements in population).

The second important feature is the *average clustering coefficient*, which for the entire network, is calculated from every single local clustering coefficient for each node. The clustering coefficient of a node shows how concentrated the neighborhood of that node is. It is possible to assume that such a feature can show the population diversity, its compactness or tendency to form heterogeneous subgroups (subpopulations).

SOCIAL NETWORKS INSIDE HEURISTICS

In this research, the complex network approach is utilized to show the linkage between different individuals in the population. Each individual in the population can be taken as a node in the complex network graph, where its links specify the successful exchange of information in the population.

Since the internal dynamics and principles are different for evolutionary (DE) and swarm-based algorithms (PSO, FA, and FWA), several different approaches for capturing the population dynamics have been developed and tested.

In the case of the DE algorithm, an Adjacency Graph was used. In each generation, the node is only active for the successful transfer of information, i.e., if the individual is successful in generating a new better individual who is accepted for the next generation of the population. If the trial vector created from three randomly selected individuals (DE/Rand/1/Bin) is better than the active individual, one establishes the connections between the newly created individual and the three sources; otherwise, no connections are recorded in the Adjacency Matrix.

Although the Firefly algorithm is a swarm type, the situation here is very similar to the evolutionary algorithms. To create a network, we decided to visualize every firefly as a node. The connection between nodes is plotted for every successful interaction between fireflies. Successful interaction is defined as such interaction where one of the individuals gets improved. In the case of Firefly algorithm, it is when firefly flies towards another and improves own brightness.

For the PSO algorithm, the main interest is in the communications that lead to population quality improvement. Therefore, only communication leading to improvement of the particles personal best (pBest) was tracked. Details and several different workflows are given later in this paper.

The last studied algorithm (FWA) is the original representative of *random search/local search engine* type algorithm. We have shown, that even for this type, it is possible to develop a scheme for capturing the communication in the form of a graph. An interesting phenomenon has been discovered. The network seems to have a lack of any other usable information, besides the ability to identify the surface type of optimized function.

EXPERIMENT DESIGN

A simple Schwefel's Test function was used in this experimental research for the generation of a complex network.

Experiments were performed, and the data were analyzed and visualized using the *Wolfram Mathematica SW* suite. Within the scope of this research, only one type of experiment was performed. It utilizes the maximum number of generations fixed at 100 with a population size of $NP = 50$ for DE and 30 for swarm algorithms, due to the better clarity of visualizations and differences between evolutionary and swarm systems. Other parameters were set up exactly as recommended by literature. Since only one run of heuristic algorithms was executed for each particular case, no statistical results and comparisons are given here.

VISUALIZATIONS FOF DE

The visualizations of complex networks are depicted in Figures 1 - 3 containing Adjacency Graphs for the selected case-studies - snapshots (beginning of the optimization process, middle part and the end of simulation). Figures 4 – 6 are depicting the corresponding community plots.

The *Degree Centrality* value is highlighted by the size of the node (red color). Analysis of CN from DE algorithm can be found in (Skanderova and Fabian 2015; Skanderova et al. 2017; Viktorin et al. 2016; Senkerik et al. 2016; Viktorin et al. 2017).

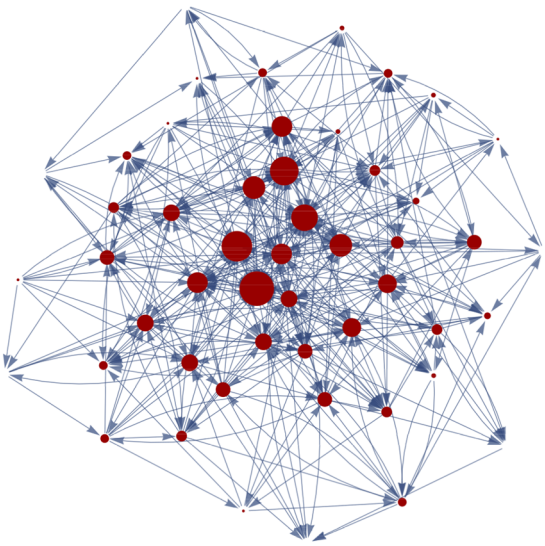

Fig. 1: CN for DE: the snapshot No.1. – the first 20 iterations.

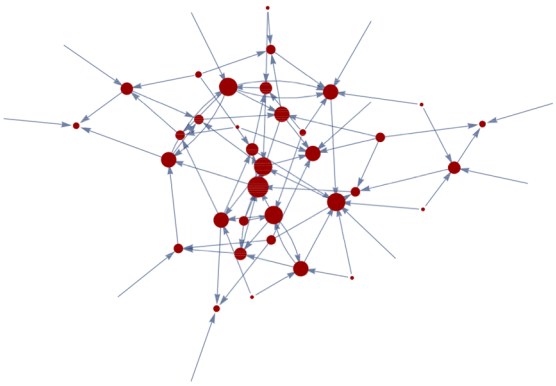

Fig. 2: CN for DE: the snapshot No.2. – the middle 20 iterations.

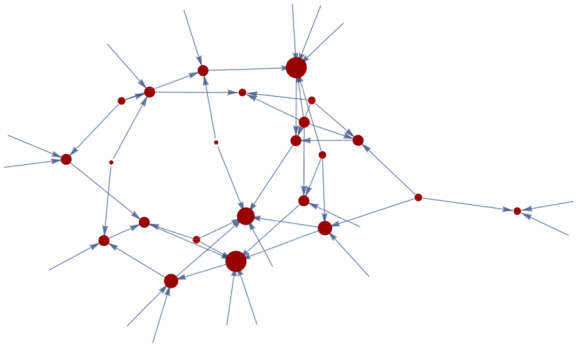

Fig. 3: CN for DE: the snapshot No.3. – the last 20 iterations.

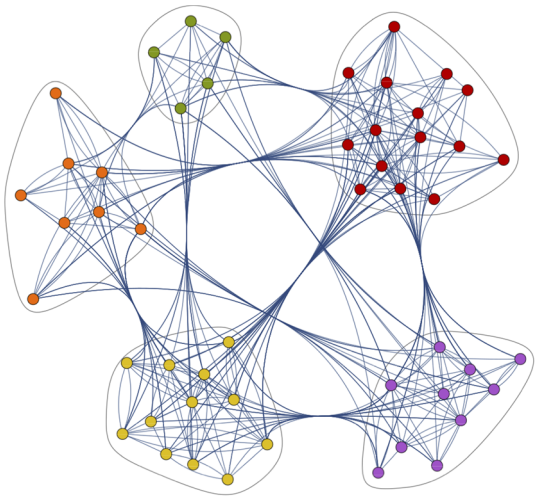

Fig. 4: Community plot for DE: the snapshot No.1. – the first 20 iterations.

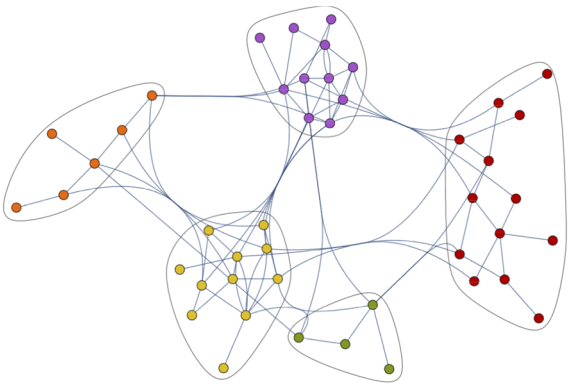

Fig. 5: Community plot for DE: the snapshot No.2. – the middle 20 iterations.

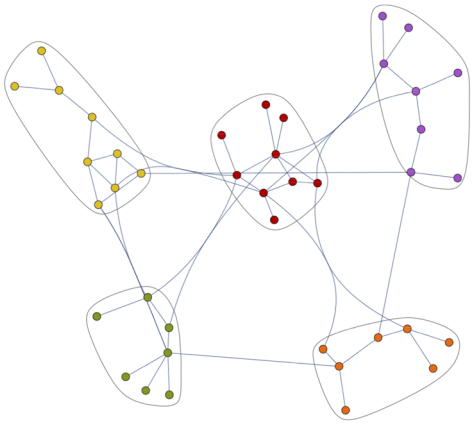

Fig. 6: Community plot for DE: the snapshot No.3. – the last 20 iterations.

VISUALIZATIONS FOR FA

In the case of FA, the connection is created, when firefly flies towards another and improves own brightness. This leads to network presented in Figure 7 (Janostik et al. 2016b). Duplicate connections were omitted.

Since across multiple iterations of the algorithm there may provide multiple connections between nodes we decided to improve upon the design by weighting the connections. If there is a connection between the firefly A and B, it starts with weight 1. If in another iteration there is another successful interaction between the firefly A and B, a new connection is not created, but the weight of the existing connection is incremented by 1. At the end of evolution, the weight is normalized. If the firefly gets improved by another in every iteration, at the end of the evolution their connection will have weight 1. If it never gets improved their connection will have weight 0. In Figure 8 we can see a network where connections have their weights visually distinguished. In the top left corner, we can see one dominant firefly which improved entire population more than 70% of iterations (blue lines). On the bottom right side we can observe few fireflies which took part in the improvement of the population only less than 30% of iterations (red lines). Also from the network, we can see that most of the fireflies improved one another only in between 30% and 70% of iterations.

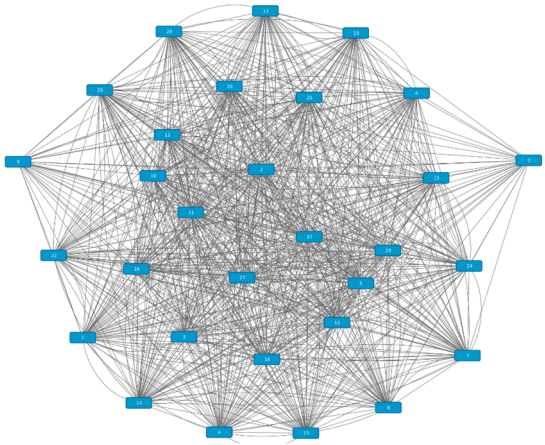

Fig. 7 Basic weighted oriented network for a population size 30 after 100 iterations.

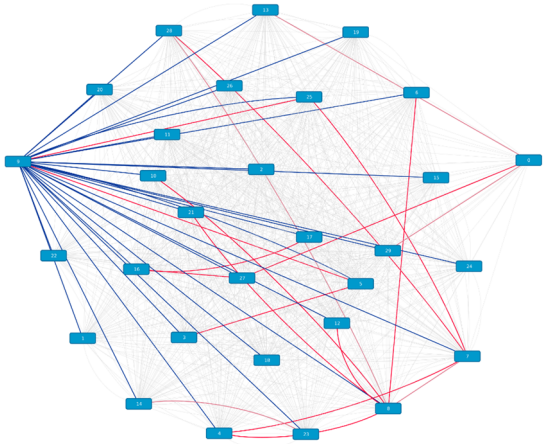

Fig. 8: a Basic weighted oriented network for a population of size 30 after 100 iterations with visually highlighted weights.

VISUALIZATIONS FOR PSO

The complex network for all iterations of the PSO algorithm that was created is depicted in Figure 9 (Pluhacek et al. 2016a). Nodes of a similar color represent particles with the same ID, and throughout different iterations. All links are from a particle that triggered the *gBest* update to a particle that has improved - based on that *gBest*. The nodes' code numbers represent a particle ID and its current iteration. This way, it is possible to precisely track the development of the network and the communication that occurs within the swarm. To be more precise, from a particular cluster, it can be observed that a single *gBest* update led to the improvement of multiple particles in different iterations. To capture the density of communication (Pluhacek et al. 2017a), the nodes in the network represent the particles in different time points (Particle ID with iteration code). This means that the theoretical maximal number of nodes in the network is the number of particles times the number of iterations. However, a new node in the network is created only when a particle manages to find a new personal best solution (*pBest*). When a node is created, two links are also created. The first link is between the newly created node and the previous node with the same particle ID (but different iteration code). This represents the information from *pBest*. Similarly the information from *gBest* represented by a link between the newly created node and a node that represents the last update of *gBest*. In the network visualizations (Figure 10) a color coding is used to differentiate the phases of the run as a percentage of the final number of cost functions evaluations (CFE). (The first 20% of CFE are represented by red color, magenta represents the 20-40% of CFE, green is the 40-60% CFE., 60-80% CFE is represented by yellow color and finally, the 80-100% CFE is represented as cyan). Such a representation can reveal the relations between the density of communication and convergence speed of the PSO.

Alternatively, it is possible to construct an Adjacency Graph and to benefit from its statistical features - as with the DE/FA case. The link is created between the particle that triggered the last *gBest* update and the particle that triggers a new *gBest* update. The self-loops (when a new *gBest* is found by the same particle as the previous *gBest*), are omitted. More studies aimed at PSO and CN framework are in (Pluhacek et al. 2016b; Pluhacek et al. 2017b).

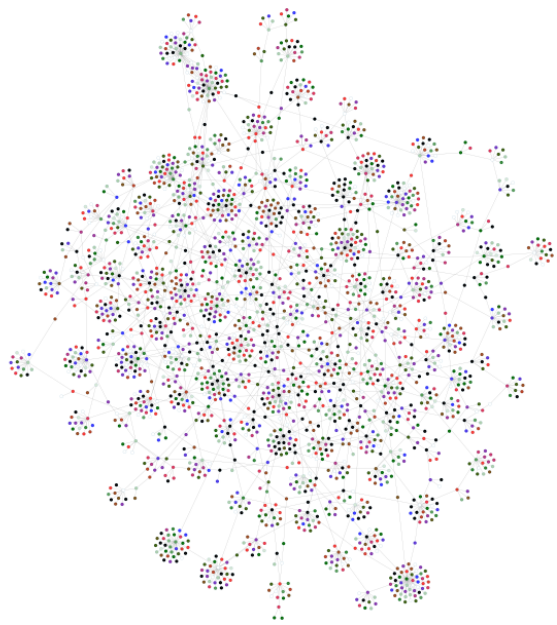

Fig. 9 PSO Dynamic as a Complex Network – Complete view (clusters).

VISUALIZATIONS FOR FWA

The network is created as a history of contributions. In each iteration, there are *NP* fireworks. These fireworks create *K* sparks. Some of these sparks are transferred into a new iteration as new fireworks. Fireworks are then represented as the nodes in the network. These nodes are labeled 1…*NP* for each iteration. The nodes (fireworks) are sorted by their fitness values before labeling so that the best node (smallest fitness value) gets number 1 and the worst node gets number *NP*. The edge between nodes represents spark that creates a new firework in next iteration. The initial node of the edge represents the firework from which the spark is created. The terminal node is the firework in the next iteration created by the spark. With that rule, the initial node from *t* iteration can have from 0 to *NP* edges, and the terminal node can only have one edge as input. The example is depicted in Figure 11 (Kadavy et al. 2017).

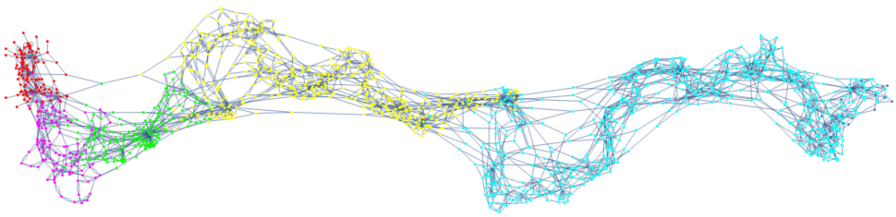

Fig. 10 PSO Dynamic as a Complex Network – Complete view (density of communication).

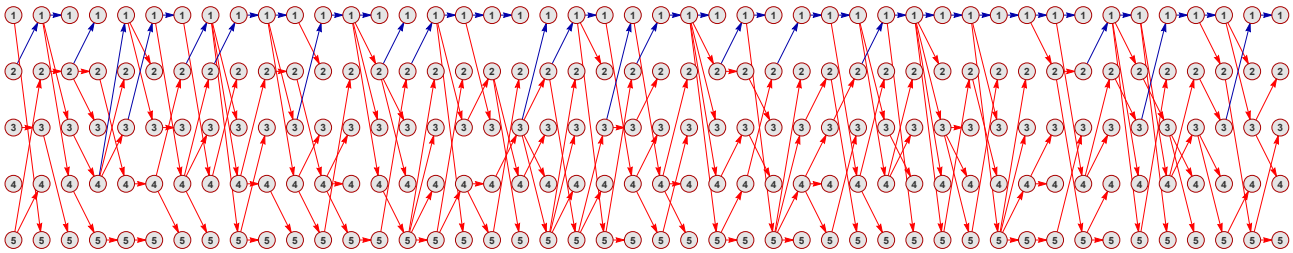

Fig. 11: FWA dynamics as a network - Blue edges indicate the spark with the best fitness function. The blue edge direction can only be towards the node number one. The first iteration is on the left side of the figure, and the last iteration is on the right side.

CONCLUSION

This work was aimed at the experimental investigation of the hybridization of a complex network framework using DE, PSO, FA and FWA algorithms. The population was visualized as an evolving complex network, which exhibits non-trivial features. These features provided a clear description of the population during evaluation.

The graphical and numerical data presented herein has fully manifested the influence of either time frame selection or type of construction to the features of the complex network. The findings can be summarized as follows:

The building of the Network: Since there is a direct link between parent solutions and offspring in the evolutionary algorithms, this information is used to build a complex network. In the case of swarm algorithms, the situation is a bit more difficult. It depends on the inner swarm mechanisms, but mostly, it is possible to capture the communications within the swarm during the updating of the information - based on the points of attraction. Several possible approaches are described herein, resulting in different graph visualizations and possible subsequent analyses.

Complex Network Features: A complex network created for evolutionary algorithms contains direct information about the selection of individuals and their success; therefore, many network features can be used for controlling a population during an EA run. At the beginning of the optimization process, intensive communication occurs (Figure 1). Later, hubs (centralities) and clusters are created (Figure 2), and it is possible to use such information either for the injection or replacement of individuals or to modify/alternate the evolutionary strategy. In the case of swarm algorithms, the communication dynamics are captured - thus the level of particle performance (usefulness) can be calculated, or some sub-clusters and centralities of such communication can also be identified - depending on the technique used for the transformation of swarm dynamics into the network.

Other Features – Fitness Landscape: Numerous previous experiments showed that there are no significant changes in complex network features for different test functions in the case of evolutionary algorithms. The capturing of communications (swarm dynamics) is sensitive to the fitness landscape. Thus, network features can be used for the raw estimation of a fitness landscape.

In this paper, we have reviewed several different approaches for visualizations, which can be, of course, hybridized and combined for any metaheuristic techniques according to user-defined requirements, which features are important to observe. Besides the presented approaches, more have been explored for a wider portfolio of algorithms (Tomaszek and Zelinka 2016; Kromer et al. 2015; Skanderova et al. 2014).

This novel topic has brought up many new open tasks, which will be resolved in future research. Another advantage is that this complex network framework can be used almost on any metaheuristic.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project no. LO1303 (MSMT-7778/2014), further by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2018/003. This work is also based upon support by COST (European Cooperation in Science & Technology) under Action CA15140, Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice (ImAppNIO), and Action IC1406, High-Performance Modelling and Simulation for Big Data Applications (cHiPSet). The work was further supported by resources of A.I.Lab at the Faculty of Applied Informatics, Tomas Bata University in Zlin (ailab.fai.utb.cz).

REFERENCES

Skanderova, L., Fabian, T., Zelinka, I. (2016). Small-world hidden in differential evolution. In Evolutionary Computation (CEC), 2016 IEEE Congress on (pp. 3354-3361).

Das S., Mullick S.S., Suganthan P. (2016) Recent advances in differential evolution – An updated survey, Swarm and Evolutionary Computation, vol. 27, pp. 1–30.

Engelbrecht A (2010) Heterogeneous Particle Swarm Optimization. In: Dorigo M, Birattari M, Di Caro G et al. (eds) Swarm Intelligence, vol 6234. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp 191-202.

Zelinka, I. (2016). SOMA—Self-organizing Migrating Algorithm. In Self-Organizing Migrating Algorithm (pp. 3-49). Springer International Publishing.

Karaboga, D. Basturk, B. (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of Global Optimization, 39 (3), pp. 459-471.

Fister I., Fister I. Jr., Yang X.S, Brest J., (2013) A comprehensive review of firefly algorithms, Swarm and Evolutionary Computation, Volume 13, Pages 34-46.

Zelinka I, Davendra D, Lampinen J, Senkerik R, Pluhacek M (2014) Evolutionary algorithms dynamics and its hidden complex network structures. In: Evolutionary Computation (CEC), 2014 IEEE Congress on, pp 3246-3251.

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

Skanderova, L., Fabian, T. (2015) Differential evolution dynamics analysis by complex networks. Soft Computing:1-15.

Metlicka, M., Davendra, D. (2015) Ensemble centralities based adaptive Artificial Bee algorithm. In: Evolutionary Computation (CEC), 2015 IEEE Congress on, pp 3370-3376.

Gajdos P, Kromer P, Zelinka I (2015) Network Visualization of Population Dynamics in the Differential Evolution. In: Computational Intelligence, 2015 IEEE Symposium Series on, pp 1522-1528.

Janostik J, Pluhacek M, Senkerik R, Zelinka I (2016a) Particle Swarm Optimizer with Diversity Measure Based on Swarm Representation in Complex Network. In: Abraham A, Wegrzyn-Wolska K, Hassanien EA, Snasel V, Alimi MA (eds) Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015. Springer International Publishing, Cham, pp 561-569.

Tan Y., Zhu Y. (2010) Fireworks Algorithm for Optimization. In: Tan Y., Shi Y., Tan K.C. (eds) Advances in Swarm Intelligence. ICSI 2010. Lecture Notes in Computer Science, vol 6145. Springer, Berlin, Heidelberg

Yang, X.S.: Nature-inspired metaheuristic algorithms. Luniver Press, Frome, U.K. (2010).

Skanderova, L., Fabian, T., Zelinka, I. (2017). Differential Evolution Dynamics Modeled by Longitudinal Social Network. Journal of Intelligent Systems, 26(3), 523-529.

Viktorin, A., Pluhacek, M., Senkerik, R. (2016). Network based linear population size reduction in SHADE. In Intelligent Networking and Collaborative Systems (INCoS), 2016 International Conference on (pp. 86-93).

Senkerik, R., Viktorin, A., Pluhacek, M., Janostik, J., Davendra, D. (2016). On the influence of different randomization and complex network analysis for differential evolution. In Evolutionary Computation (CEC), 2016 IEEE Congress on (pp. 3346-3353).

Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T. (2017). Towards better population sizing for differential evolution through active population analysis with complex network. In Conference on Complex, Intelligent, and Software Intensive Systems (pp. 225-235).

Janostik, J., Pluhacek, M., Senkerik, R., Zelinka, I., Spacek, F. (2016b). Capturing inner dynamics of firefly algorithm in complex network—initial study. In Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015 (pp. 571-577).

Pluhacek, M., Janostik, J., Senkerik, R., Zelinka, I., Davendra, D. (2016a). PSO as complex network—capturing the inner dynamics—initial study. In Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015 (pp. 551-559).

Pluhacek, M., Senkerik, R., Janostik, A. V. J., Davendra, D. (2016b). Complex network analysis in PSO as an fitness landscape classifier. In Evolutionary Computation (CEC), 2016 IEEE Congress on (pp. 3332-3337).

Pluhacek, M., Senkerik, R., Viktorin, A., Kadavy, T. (2017a). Uncovering communication density in PSO using complex network. In Proceedings-31st European Conference on Modelling and Simulation, ECMS 2017. European Council for Modelling and Simulation.

Pluhacek, M., Viktorin, A., Senkerik, R., Kadavy, T., Zelinka, I. (2017b). PSO with Partial Population Restart Based on Complex Network Analysis. In International Conference on Hybrid Artificial Intelligence Systems (pp. 183-192).

Kadavy, T., Pluhacek, M., Viktorin, A., Senkerik, R. (2017). Firework algorithm dynamics simulated and analyzed with the aid of complex network. In Proceedings-31st European Conference on Modelling and Simulation, ECMS 2017. European Council for Modelling and Simulation.

Tomaszek, L., Zelinka, I. (2016). On performance improvement of the SOMA swarm based algorithm and its complex network duality. In Evolutionary Computation (CEC), IEEE Congress on (pp. 4494-4500).

Krömer, P., Gajdos, P., Zelinka, I. (2015). Towards a Network Interpretation of Agent Interaction in Ant Colony Optimization. In Computational Intelligence, 2015 IEEE Symposium Series on (pp. 1126-1132).

Skanderova, L., Zelinka, I., Saloun, P. (2014). Complex Network Construction Based on SOMA: Vertices In-Degree Reliance on Fitness Value Evolution. In ISCS 2013: Interdisciplinary Symposium on Complex Systems (pp. 291-297). Springer Berlin Heidelberg.