

Chapter 2

Evolutionary Dynamics and Its Network Visualization - Selected Examples

Orkhan Yarakhmedov, Victor Polyakh, Ivan Chernogorov
and Ivan Zelinka

Abstract In this chapter is demonstrated, on selected algorithms, how can be converted dynamics of the evolutionary process into a network. Selected algorithms are a self-organizing migrating algorithm, differential evolution, particle swarm, artificial bee colony and ant colony optimization. The main ideas and steps are discussed here, for more detailed study and understanding references to original research papers are throughout the text. The aim of this chapter is to show principles of conversion with attention on the fact that there is no universal guide on how to do conversion - it is a matter of creative approach and algorithm knowledge.

2.1 Introduction

In the basic visualization (see the previous chapter) is proposed that vertices of the network be individuals and number of edges (or its strength) between them only increase by an integer number. It is also possible to set decrement of edges according to various criteria. For example when parents - individuals do not create better offsprings, then the number of edges between them decrease. Each evolutionary algorithm is running under different nature, so there is no universal guide how to

O. Yarakhmedov (✉) · V. Polyakh · I. Chernogorov
Software of Computer Engineering and Automated Systems Department,
Information Technology and Computer Science Faculty,
Don State Technical University, Rostov, Russia
e-mail: orhashka@gmail.com

V. Polyakh
e-mail: silvervpolyah@gmail.com

I. Chernogorov
e-mail: hintaivr@gmail.com

I. Zelinka
Faculty of Electrical Engineering and Computer Science,
Department of Computer Science, VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: ivan.zelinka@vsb.cz

© Springer-Verlag GmbH Germany 2018
I. Zelinka and G. Chen (eds.), *Evolutionary Algorithms, Swarm Dynamics
and Complex Networks*, Emergence, Complexity and Computation 26,
https://doi.org/10.1007/978-3-662-55663-4_2

convert dynamics of studied EA to complex network and has to be developed for each specific algorithm. For example, SOMA, see [1]), is in each migration loop winner only one individual, so-called Leader, and other individuals establish a connection with this individual. On the other side, DE creates for selected parent individual, using randomly selected another three individuals (DERand1Bin) new offspring which are understood here as the movement of selected parent individual to the new position. If fitness is better, then this selected parent individual get new connections from those randomly selected another three individuals. Decreasing of existing edges is done in a similar inverse way. Complex networks, generated from EAs dynamics, can be done, for example, in following scenarios:

- **Scenario 1 (increasing mode):** number of established edges increase only for individuals whose fitness was improved.
- **Scenario 2 (real mode):** number of edges between vertices is decreasing according to various scenarios:
 - **Scenario 2a:** if edge between individual A and B exist, then is decreased by 1, if not exist, then decrease is not done, i.e. minimal value that can be reached is 0
 - **Scenario 2b:** if edge between individual A and B exist, then is decreased by 1 even if not exist, i.e. minimal value that can be negative.
- **Scenario 3:** number of edges between vertices is increasing and decreasing according to various scenarios that can involve fitness and another attribute of EA dynamics

For more details about results and visualizations, please see [2].

In this chapter are reported examples of the self-organizing migrating algorithm, differential evolution, particle swarm, an artificial bee colony and ant colony optimization and conversion of its dynamics into the network. Algorithms have been tested on various test function (to reveal its complex networks dynamics) with a constant level of test function dimensionality (i.e. individual length) and a different number of generations (migrations) in all used algorithms. All data has been processed graphically (see [2], etc.) alongside calculations of basic network properties. The emergence of network structure behind evolutionary dynamics depends on many factors. However, some special versions of used algorithms did not show complex network structure despite the fact that the number of generations was quite large. Pictures in this chapter have been generated to visualize all important data and relations, see also [1–21]. Another part of our experiments and results also show the use of parallel evolutionary algorithms and its conversion and visualization as complex networks.

Based on mentioned principles and algorithms, we believe that there is no universal approach, but rather a personal one, based on the knowledge of algorithm principle. Lets take as an example DE, e.g. DERand1Bin in which each individual is selected in each generation to be a parent. Thus in DE, we have recorded only those

individuals-parents, which have been replaced by better offspring (like vertex with added connections). In the DE class of algorithms, we have omitted the philosophy that a bad parent is replaced by a better offspring, but accepted philosophical interpretation, which individual (worse parent) is moving to the better position (better offspring in original DE philosophy). Thus no vertex (individual) has to be either destroyed or replaced in the philosophical point of view. If, for example, DERand1Bin has a parent replaced by offspring, then it was considered as an activation (new additional links, edges) of vertex-worse parent from three another vertices (randomly selected individuals, see [4, 15–17]). In fact, such general approach can also be used in another algorithm. If interaction amongst N individuals leads to the improvement of another arbitrary one, say M_{th} , then it is equivalent to the situation that N vertices give incoming edges to the M_{th} vertex. Moreover, vice versa. Edges can be weighted by integer numbers ($+1$ = successive support of the M_{th} vertex, -1 = nonsuccessive support) or simply by differences in fitness before interaction and after the interaction. Again, this very general idea has to be adopted for each used algorithm. As reported in [6–9], we have tried this conversion for SOMA, DE, PSO, ABC (Artificial Bee Colony) and GA. Complex networks can be then visualized as in figures below. From figures, it is visible that different algorithm dynamics produce different complex networks structure. Another visualizations can be found in [6–9].

2.2 Interpretation

Visualized and analyzed networks have multiple edges that can be understood like the weight of the single edge. Attributes of proposed analysis are represented by subnetwork colors and vertices size in the network. Networks can be analyzed from the different point of views. The most important is an interpretation of what network, its structure, and dynamics represent. Our proposed interpretation, based on terms and command from Wolfram *Mathematica* is following:

1. Adjacency graph, see Fig. 2.1.

Meaning: Graph with vertices and oriented edges.

Interpretation: Visualization of evolutionary dynamics in the form of so called graph. Each vertex represent one individual in the population and each edge (oriented of course) represent successful offspring creation (i.e. fitness improvement of active parent in this philosophy) between parents connected by that edge.

2. Graph partition, see Fig. 2.2.

Meaning: Graph partition finds a partition of vertices such that the number of edges having endpoints in different parts is minimized. For a weighted graph, graph partition finds a partition such that the sum of edge weights for edges having endpoints in different parts is minimized.

Interpretation: Individuals in a population are separated into “groups” according to their interactions with another individual, based on their success in active individual fitness improvements. “Endpoints” can be understood like successful participation of selected individuals in active individual fitness. On Fig. 2.2 is partition visualized by colors. This analysis gives a view of population structure and shows the set of individuals that got or donate oriented edges (support from/to) the same group of individuals. Based on some connections or weights of edge, it can be analyzed what part of the population was the most important in the evolutionary dynamics for given case.

3. **Degree centrality**, see Fig. 2.3.

Meaning: Degree centrality of g gives a list of vertex degrees for the vertices in the underlying simple graph of g . Degree centrality will give high centralities to vertices that have high vertex degrees. The vertex degree for a vertex v is the number of edges incident to v . For a directed graph, the in-degree is the number of incoming edges, and the out-degree is the number of outgoing edges. For an undirected graph, in-degree and out-degree coincide.

Interpretation: Degree centrality shows how many incoming (support from individuals), or out-coming (support to individuals) edges vertex - individual under study has. This quantity can be related to the progress of the evolutionary search and used to make the conclusion of what set of individuals has maximally contributed to that. On Fig. 2.3 are individuals sized according to that degree.

4. **Community**, see Fig. 2.4.

Meaning: Community graph plot attempts to draw the vertices grouped into communities.

Interpretation: Community graph plot showing the individuals grouped into communities. Communities (with border are individuals that communicate amongst themselves (higher density of edges in the community, multi-edges are not visualized here, rather than between communities) and community are then joined by connections that are “one-way” and shows the flow of information between communities). This kind of visualization can be interesting also in the case of parallel EAs, where islands of subpopulations are formed.

Beside standard analysis of networks, it can study how information has been spread throughout population during generations (i.e. network) as a flow of influence. In Fig. 2.5 is captured how was influenced individual 50 (vertex 50) by individual No. 1 (vertex 1). A set of possible paths is depicted by thick blue color there. It is also clear that a part of the population has been influenced too. This is one example of many. The whole tool of network analysis can be applied on networks generated by evolutionary dynamics. For a more comprehensive overview and interpretations of complex network properties and EAs parameters and structure it is recommended to read for example [22], but also [1–21].

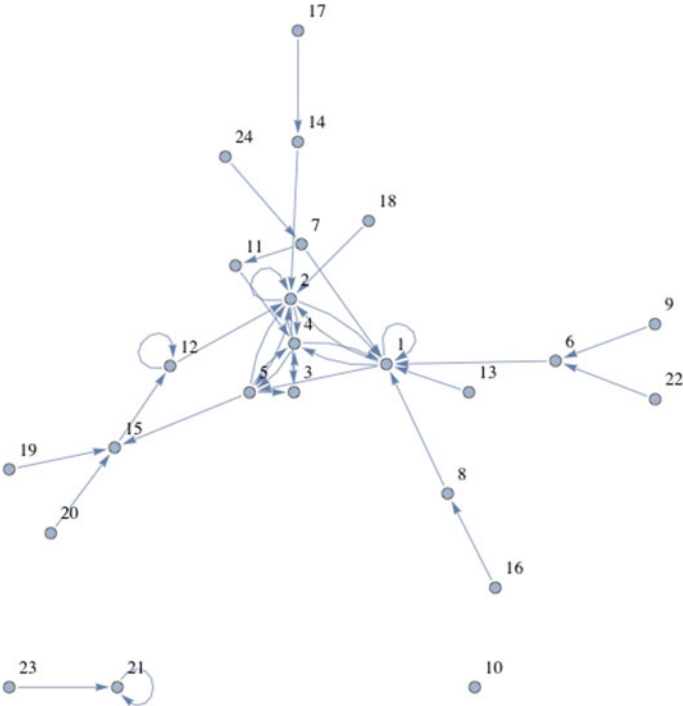


Fig. 2.1 Adjacency graph

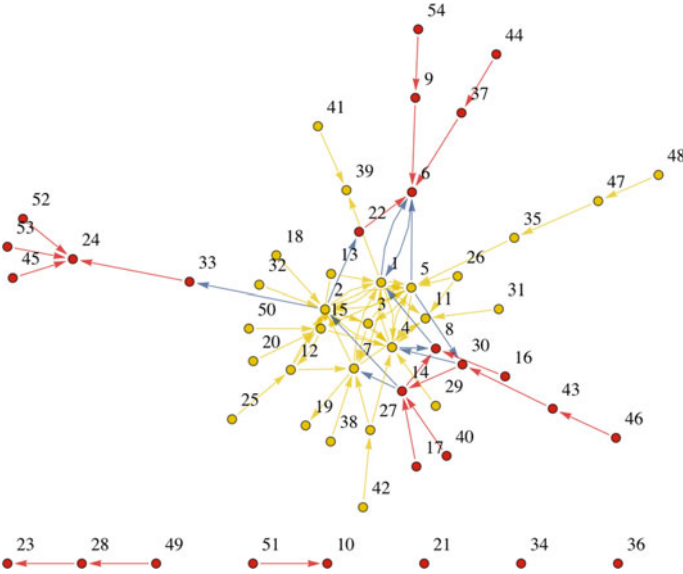


Fig. 2.2 Graph partition

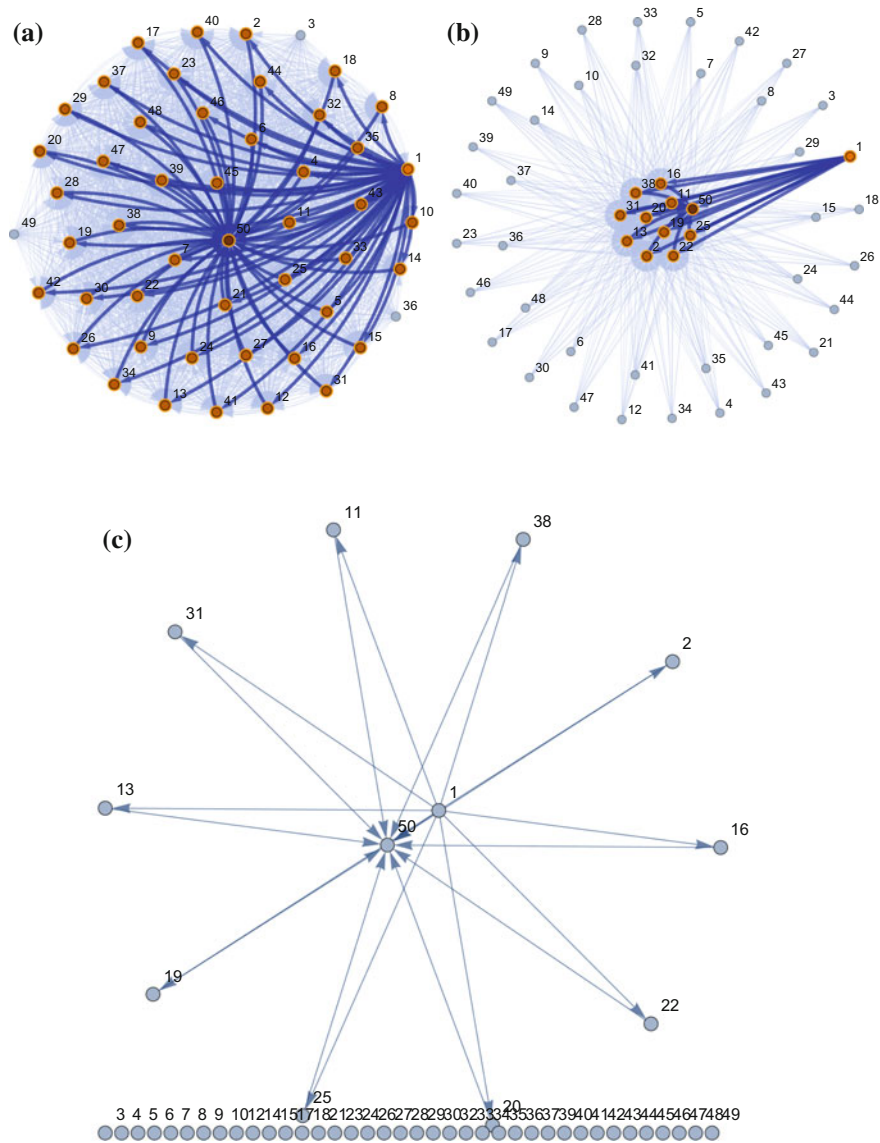


Fig. 2.5 Example of a flow-influence from individual 1 to 50 in SOMA algorithm in **a** 200 generations, **b** 20 generations, **c** extracted from **(b)**

2.3 Algorithm Networks

2.3.1 Self-organizing Migrating Algorithm

SOMA is a stochastic optimization algorithm, modeled based on the social behavior of competitive-cooperating individuals [1]. It was chosen because it has been proved that this algorithm has the ability to converge towards the global optimum [1]. SOMA works on a population of candidate solutions in loops, called migration loops. The population is initialized by being randomly and uniformly distributed over the search space at the beginning of the search. In each loop, the population is evaluated, and the solution with the lowest cost value becomes the leader. Apart from the leader, in one migration loop, all individuals will traverse the searched space in the direction of the leader. Mutation, the random perturbation of individuals, is an important operation for evolutionary strategies. It ensures the diversity among all the individuals and it also provides a means to restore lost information in a population. Mutation is different in SOMA as compared with other evolutionary strategies. SOMA uses a parameter called PRT to achieve perturbations. This parameter has the same effect for SOMA as mutation for GA. The novelty of this approach lies in that the PRT vector is created before an individual starts its journey over the search space. The PRT vector defines the final movement of an active individual in the search space. The randomly generated binary perturbation vector controls the permissible dimensions for an individual. If an element of the perturbation vector is set to zero, then the individual is not allowed to change its position in the corresponding dimension. An individual will travel over a certain distance (called the PathLength) towards the leader in a finite number of steps in the defined length. If the PathLength is chosen to be greater than one, then the individual will overshoot the leader. This path is perturbed randomly. More about SOMA, see in [1, 5]. Networks generated by SOMA and Scenario 1 is depicted in Fig. 2.6. When applied to parallel hardware, then network separated to subnetworks, weakly connected, can be expected, Fig. 2.7.

2.3.2 Differential Evolution

Differential evolution [4] is a population-based optimization method that works on real-number-coded individuals. For each individual $\mathbf{x}_{i,G}$ in the current generation G , DE generates a new trial individual $\mathbf{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\mathbf{x}_{r1,G}$ and $\mathbf{x}_{r2,G}$ to a randomly selected third individual $\mathbf{x}_{r3,G}$. The resulting individual $\mathbf{x}'_{i,G}$ is crossed-over with the original individual $\mathbf{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\mathbf{u}_{i,G+1}$, is then compared with the fitness of $\mathbf{x}_{i,G}$. If the fitness of $\mathbf{u}_{i,G+1}$ is greater than the fitness of $\mathbf{x}_{i,G}$, then $\mathbf{x}_{i,G}$ is replaced with $\mathbf{u}_{i,G+1}$; otherwise, $\mathbf{x}_{i,G}$ remains in the population as $\mathbf{x}_{i,G+1}$. DE is quite robust, fast, and effective, with a global optimization ability. It does not require the objective function be differentiable, and it works well even with

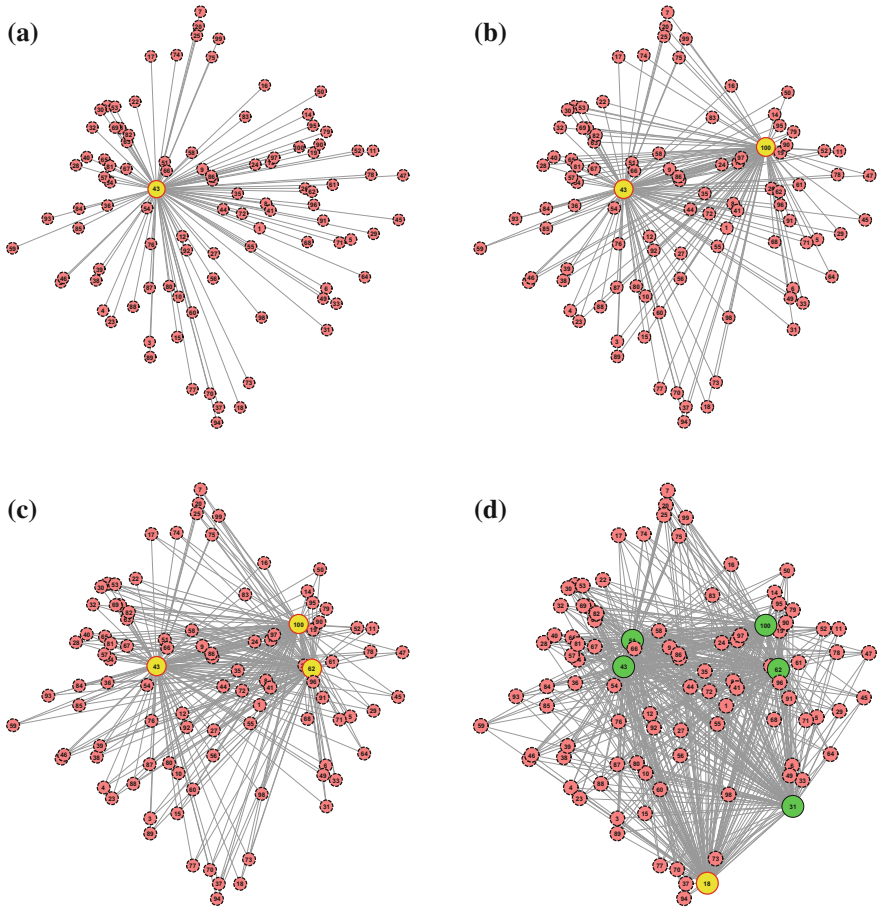


Fig. 2.6 Example of a network growth of SOMA algorithm since beginning, **a** 1st generation, **b** 2nd generation, **c** 3rd generation, **d** N th generation. The *yellow* vertex represent the best individual from the population

noisy, epistatic and time-dependent objective functions. For more about DE see in [4].

In this case of [7, 10, 11] is discussed use of the weighted clustering coefficient [13, 14] in order to improve DE performance. One of the main goals of weighted clustering coefficient use on DE is to find the way how to select the parents in the mutation step to improve DE convergence rate. In the variant DE/best/1/bin, the best individual is with the smallest (in minimizing problems) or greatest (in maximizing problems) fitness value. As it was described in former research, DE/best/1/bin does not reach the best results in the non-separable and multimodal functions. This fact led us to search for another criterion of the best individual selection. From the rules that we have presented, we know that the edges between nodes representing the

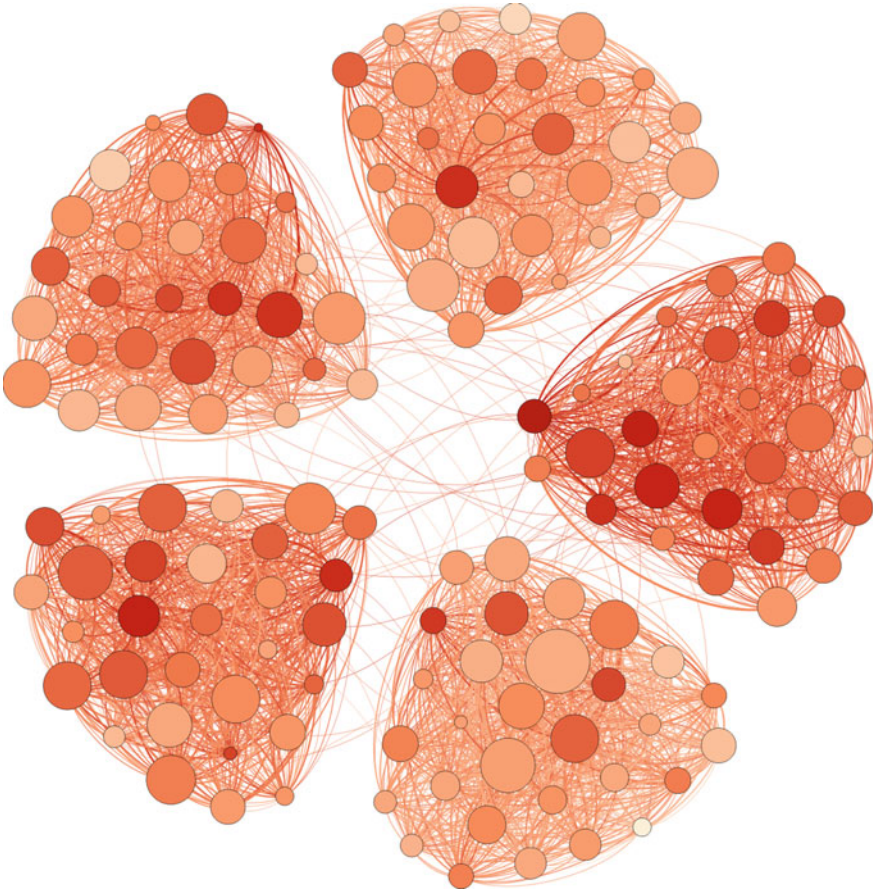


Fig. 2.7 SOMA network on parallel platform. Five subnetworks has been created during the evolutionary process

individuals are created only if the individuals represented by these nodes contributed to the population improvement. The nodes representing the individuals becoming the parents will have the greater out-degree (the number of out-going edges will be greater) more often, and the weights of the edges leading from these nodes will be higher. Such parents have the genomes of high quality, and it is important to spread these genomes. The weighted clustering coefficient is one of the most appropriate candidates to be chosen as the criterion of parents selection in the mutation step from two reasons:

1. It enables to capture spreading of the genomes from the node to another node (individual).
2. Unlike the local clustering coefficient, the weights of the edges are taken into consideration.

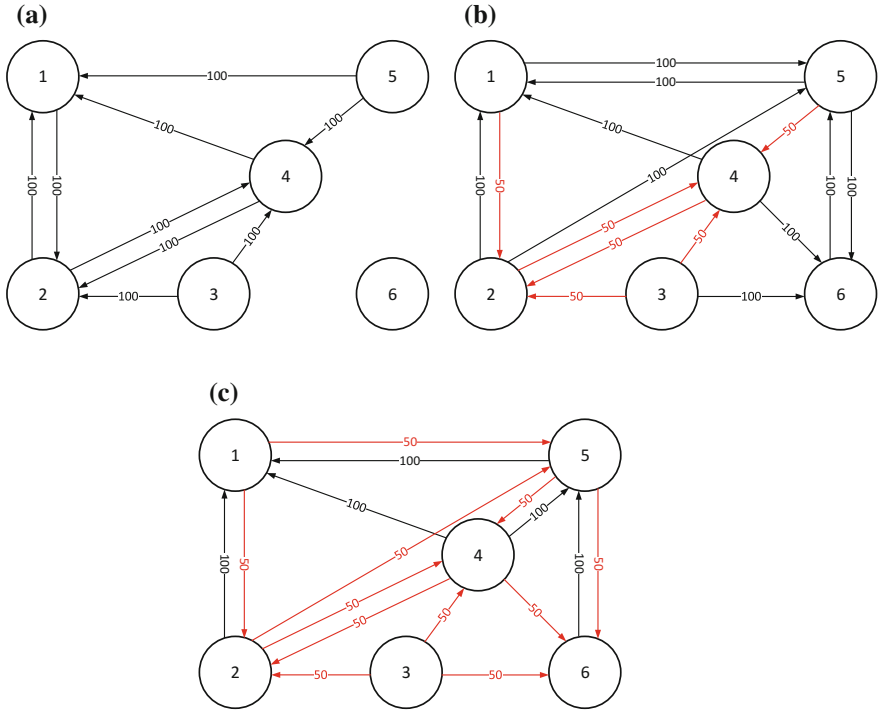


Fig. 2.8 DE network principal growth, **a** 1st generation, **b** 2nd generation, **c** 3rd generation

When the adjacency matrix A_G for the generation G is created, the weighted clustering coefficient of each node is computed. Then, for each individual the probability of selection is computed according to the following equation:

$$p_i = \frac{\tilde{C}_{i,Z}}{\sum_{j=1}^{NP} \tilde{C}_{j,Z}}, \quad (2.1)$$

where p_i denotes the probability of the i -th individual selection and $\tilde{C}_{i,Z}$ is the weighted clustering coefficient of the node representing the i -th individual. Individual represented by the node with the higher weighted clustering coefficient has, the higher probability to be selected as the parent in the mutation step. The network can be constructed as suggested in Fig. 2.8. The result can be then networked as in Fig. 2.9 or 2.10. As reported in [10, 11], this approach has improved DE's performance significantly. Similar research is reported in [7].

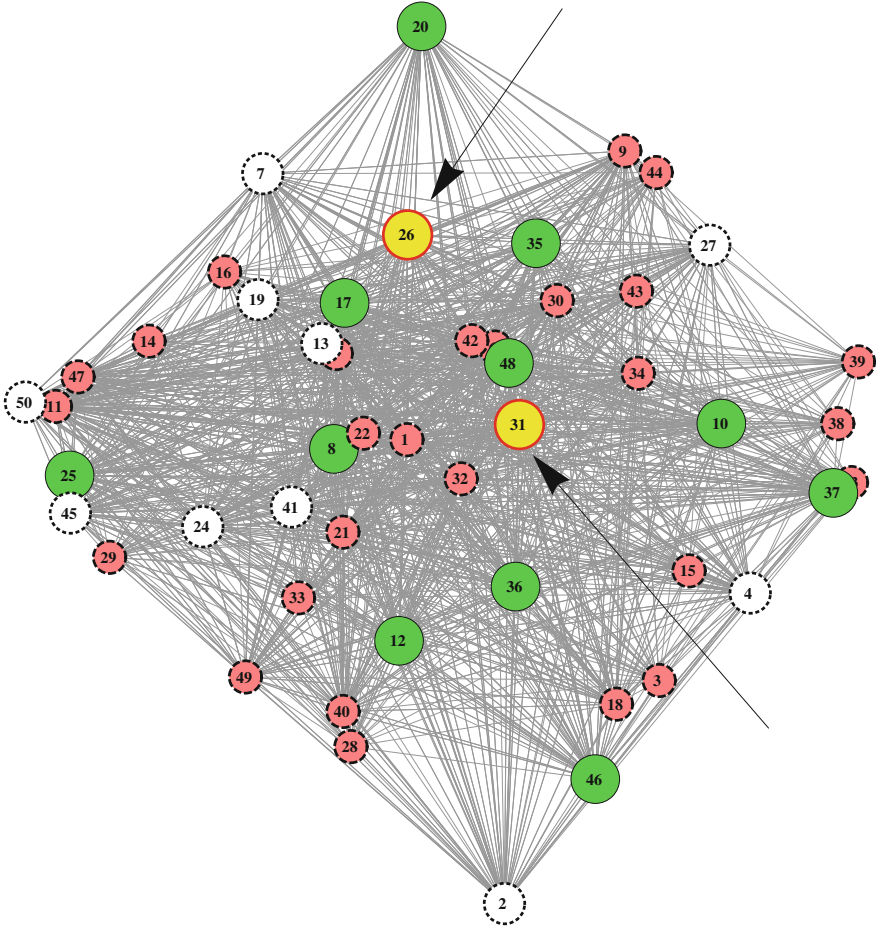


Fig. 2.9 Complex network of the DELocalToBest with two the most intensively connected vertices (individuals) ...

2.3.3 Particle Swarm

Papers [18] presents an initial proposal of methodology for converting the inner dynamics of PSO algorithm into a complex network. The motivation is in the recent trend of adaptive methods for improving the performance of evolutionary computational techniques. It seems very likely that the complex network and its statistical characteristics can be used within those adaptive approaches. The methodology described in this paper manages to put a significant amount of information about the inner dynamics of PSO algorithm into a complex network as depicted in Figs. 2.11 and 2.12, see [18–21].

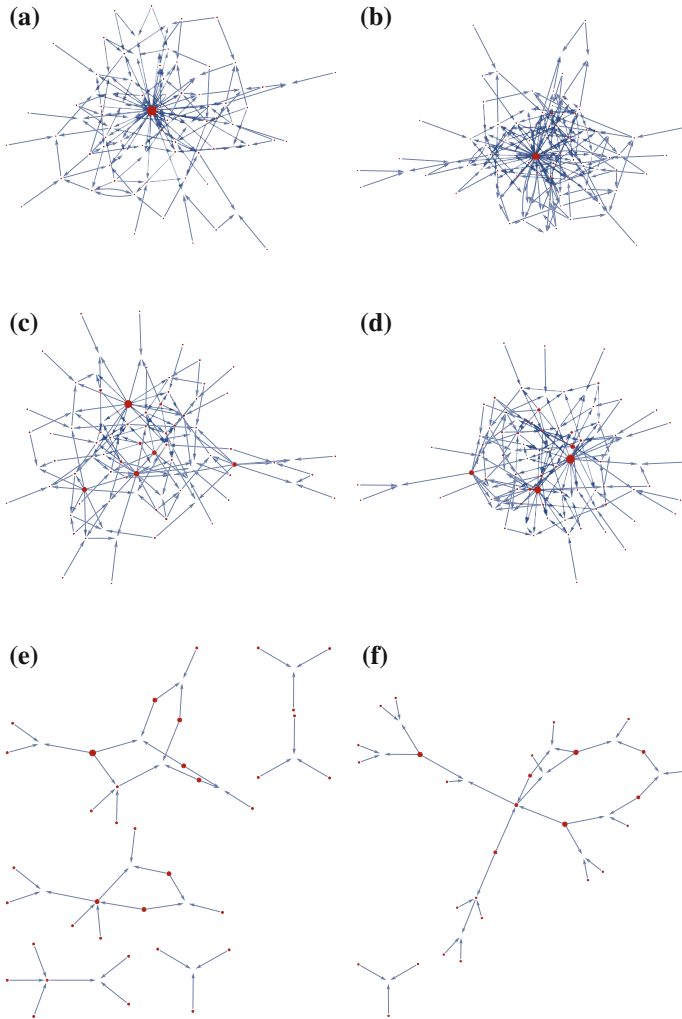


Fig. 2.10 DE network examples; **a** and **b** HSDE, **c** and **d** JADE, **e** and **f** DERand1Bin

2.3.4 Artificial Bee Colony

Proposed approach has also been tested on Artificial Bee Colony algorithm (ABC) and presented in [6, 8]. The main idea was to test ABC on scheduling problems as well as on continuous test problems. The complex network analysis was used for adaptive control of the population. The structure of the algorithm was as follows: firstly, the weighted adjacency matrix was created throughout the algorithm iterations, for some fixed number of iterations, a fraction of the total expected some iterations before algorithm termination. The complex network recorded this way is then analyzed,

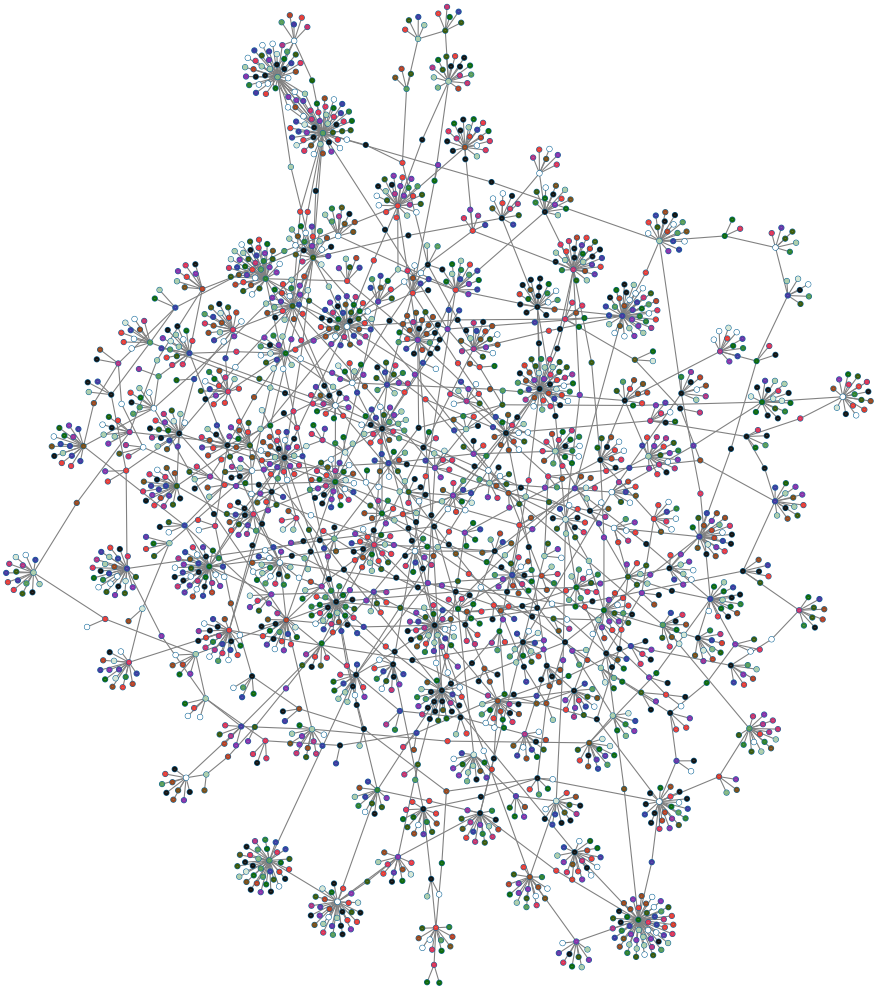


Fig. 2.11 Example of a network based on PSO. Network structure capture time development of PSO in certain time window

and this information was subsequently used to identify the nodes (solutions) that don't play a significant role in the population dynamics. In this algorithm, such nodes are replaced by the new randomly generated ones, although different schema of the replacements generation could also be used.

The measures used to identify the nodes that do not contribute significantly to the population improvement were chosen to be the three types of vertex centrality, the weighted degree centrality (strength), closeness and betweenness centrality, as described in [6, 8]. The vertices representing solutions were ranked according to these measures, and the fixed ratio of the solutions corresponding to the lowest ranking

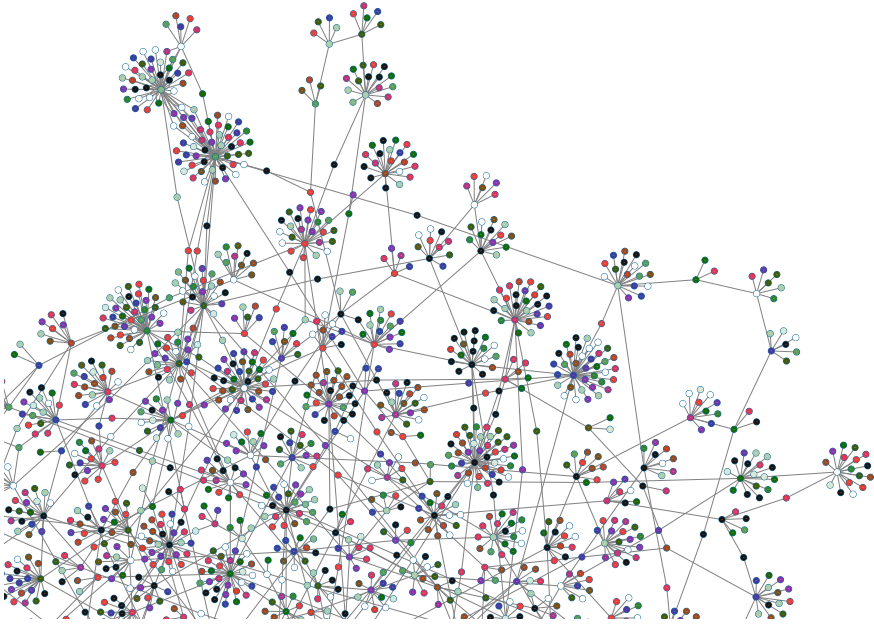


Fig. 2.12 Example of a network based on PSO. Network structure capture time development of PSO in certain time window - detail

nodes was removed and regenerated. The adjacency matrix was then reset. The entire procedure of network recording and the nodes ranking and replacement are repeated until the algorithm terminates. This concept is illustrated in Figs. 2.13, 2.14 and 2.15.

ABC was used in other 2 (so in total in 3) variants. The 2nd variant (called in [8] Adaptive ABC 2) is based on the use of three fully separated sub-populations, each with their own network (or better sub-network, if whole EAs dynamics is taken like network itself). Each of them was evaluated using different centrality measure type. After the less influential nodes are pruned, and the networks reset, the tournament selection of size two is performed to select every next-generation solution of every sub-population, choosing the better of two solutions randomly selected from all three sub-populations. In this way, the information sharing between sub-populations is ensured.

The third version, named Adaptive ABC 3 was created to explore the influence of distinct centralities on the speed of convergence. It takes the centrality measure to be used for network analysis and nodes evaluation as a parameter. The algorithm is fully described in [8].

Based on results reported in above-referenced papers, it can be stated that concept proposed here was successful and has improved ABC performance significantly [8].

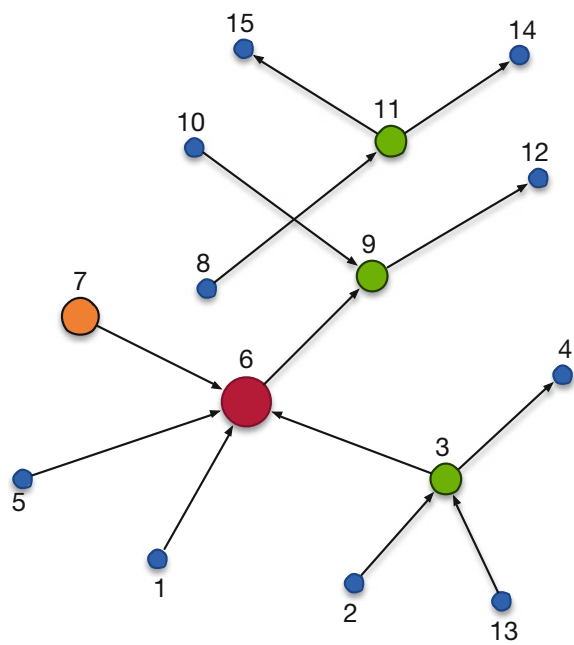


Fig. 2.13 The network with labeled nodes (individuals) ranked by centrality. The larger centrality nodes are marked in bigger size and different colors. The smallest blue nodes have the lowest centrality, the largest red node has the highest centrality value

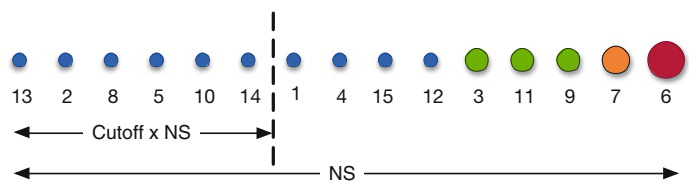
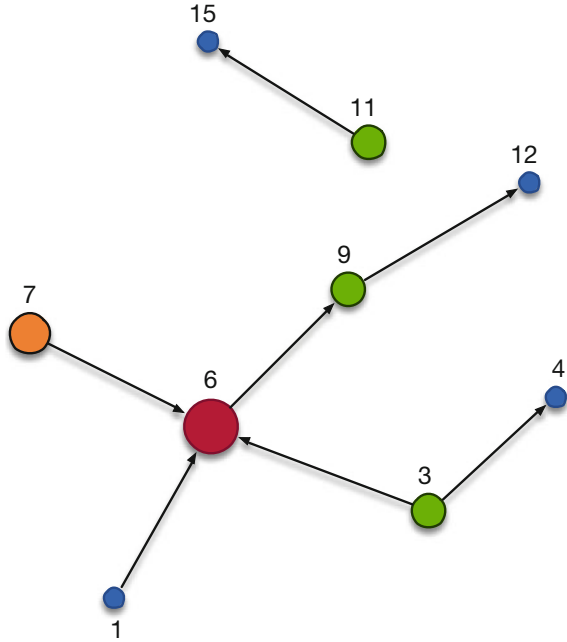


Fig. 2.14 The nodes (individuals) sorted according to their centrality score in ascending order. The first Cutoff NS nodes will be removed

2.3.5 Ant Colony Optimization

The Ant Colony Optimization (ACO) is another group of methods used in solving different optimization tasks. The ACO’s distinctive feature is that the key behavioral characteristics of real ants are simulated [23]. Commonly, ACO is mostly applied to minimize the path in graph tasks [24], but the given algorithms also show good results in other domains [25, 26]. In this chapter, we will describe the complex network construction’s features of social interaction within an ants colony. The authors study the multi-parameter benchmark tasks such as Rastrigin, Rosenbrock, Himmelblau, Schwefel26, Giunta and Ursem01 functions. Some of these functions are

Fig. 2.15 The network after the low centrality nodes (individuals) removal. The most important nodes are preserved



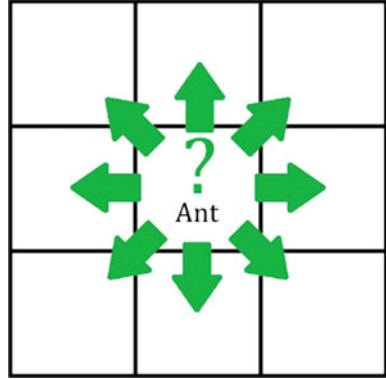
characterized by multimodality, so, it undoubtedly will affect the final structure of the network. The described method, based on the classic ACO realization, is applied for solving graph problems [26], however with some additions. Similar to the classic ACO, in this modification, we can distinguish such steps as initialization and arrangement, moving of ants, updating of pheromone and breakpoint checking. All researching space is divided into $n \times n$ fragments, and each of them associated with function value in the center and some pheromone level. The specified amount of ants is placed on each fragment. As all the fragments are equal, the size of the fragment can be calculated by the following formula:

$$m = (X_{\max} - X_{\min})/n \quad (2.2)$$

Thus, the set of fragments is defined by the matrix $B = (I_{i,j})_{i=q,j=l}^{n,n}$. When an ant is moving from fragment $I_{(i,j)}$, it is calculating the moving probabilities towards the adjacent fragments (see Fig. 2.16), by using the following formula:

$$P_{i,j}^k = \begin{cases} \frac{\tau_{i,j}^\alpha \times \eta_{i,j}^\beta}{\sum_{i,j=1}^8 \tau_{i,j}^\alpha \times \eta_{i,j}^\beta}, & \text{if } i, j \text{ available} \\ 0, & \text{if } i, j \text{ not available} \end{cases} \quad (2.3)$$

Fig. 2.16 Available fragments to move



where: i, j - fragments; k - ant number; τ - pheromone level; η - visibility- difference between two fragments function values; α, β - variable coefficients; $*$ - available if fragment both adjacent and not an obstacle.

In each iteration pheromone both increases and evaporates. Thus the amount of pheromone increment for the simulation step in the $I_{(i,j)}$ fragment is calculated by the following formula:

$$\tau_{ij}(t+1) = (1 - \rho) * \tau_{ij}(t) + \Delta \tau_{ij}(t) \quad (2.4)$$

where: $\rho \in (0; 1)$ - evaporate coefficient; $\tau_{ij}(t)$ - pheromone level in $I_{(i,j)}$ fragment; $\Delta \tau_{ij}(t)$ - increment at each iteration, is calculated by the next formula:

$$\Delta \tau_{ij}(t) = \sum_{p=1}^q (K * (f(x_{i,j}, y_{i,j})^{(k_p)} > f(x_{i+1,j+1}, y_{i+1,j+1})^{(k_p)})) \quad (2.5)$$

where K pheromone increment coefficient. Evaporation occurs because all the space of possible solutions should be re-searched.

2.3.5.1 Experiment Design

By the described algorithm and the model (see Eqs. 2.2–2.5) a software tool (ST) that implements the search of local and global extremes was developed alongside with a mechanism of analysis of ants interaction with each other. During every iteration, ants move from fragment to fragment using pheromone traces of another ant. It is the mechanism of experience exchange within the colony. You can see it step by step on Fig. 2.17.

As you can see in Fig. 2.17 all ants tend to migrate to one fragment (gray one) where the extreme is situated. In the process of building the ants are attracted to the extreme, and also use the pheromones of other ants. The first ant which is marked by

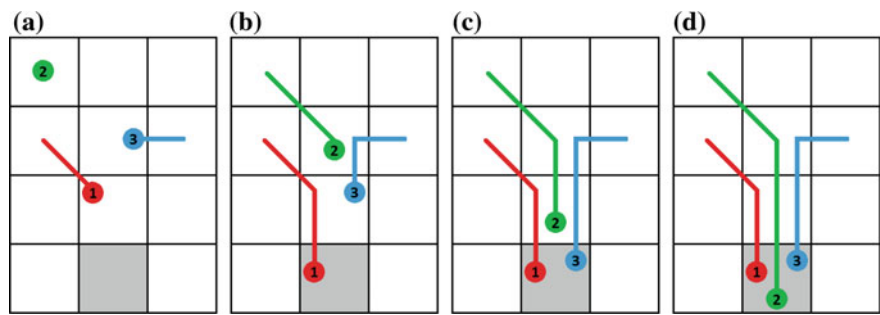
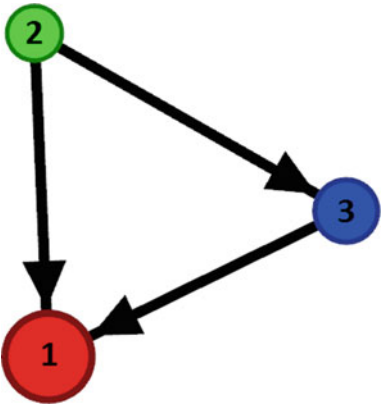


Fig. 2.17 Mechanism of experience exchange within the colony: **a** 1st step, **b** 2nd step, **c** 3rd step, **d** 4th step

Fig. 2.18 Graph of relations between ants



red color builds its route using only attraction. The third ant (the blue one) builds a route using pheromone trace of the first ant, and the second ant (green one) reaches extreme fragment using both the first and the second ant’s pheromone traces. Using this data author can create a graph of relations between ants. You can see in in Fig. 2.18.

Using this approach, the authors have analyzed the behavior of ants in the process of optimization of different test functions.

2.3.5.2 Results

All graphs described below were visualized using free software tool Gephi. For the layout, the graphs authors used force-direct Yifan Hu algorithm [27]. The first test function is Rastrigin function. The function proposed in 1974 by Leonard Rastrigin [28] demonstrates the unlimited multimodal function. The equation of the n -dimensional Rastrigin test function is:

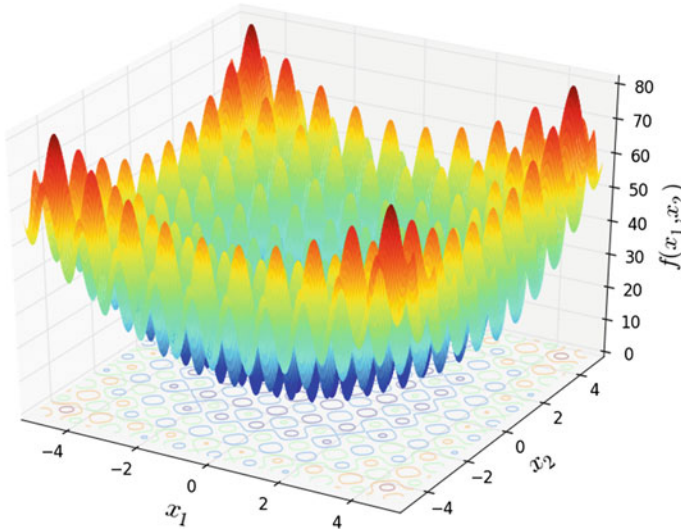


Fig. 2.19 Rastrigin function chart

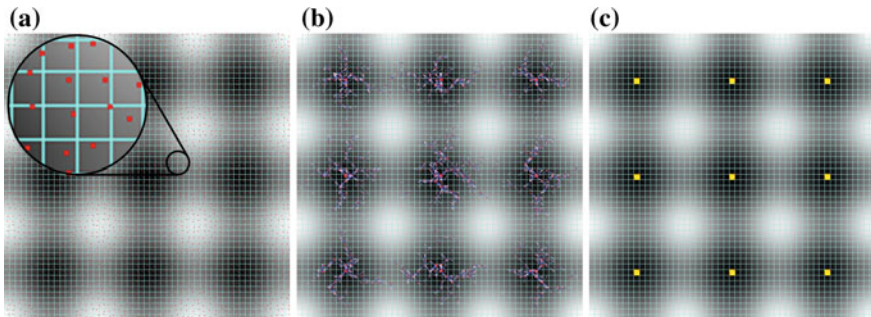


Fig. 2.20 Rastrigin function optimization process: **a** initialization step, **b** middle step, **c** final result

$$f_{Rastrigin}(x) = 10n \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad (2.6)$$

Its 2-dimensional view is represented in the Fig. 2.19.

Figure 2.20 shows the process of optimization of a Rastrigin function. The function was studied in the range $[-1.5; 1.5]$.

Figure 2.21 shows result graph. In Fig. 2.21 you can see all ants and connections between them. Obviously, complex graph consists of some clusters of vertexes - 9. The same number of extremes in the considered area. Also, it should be noted, that there are links between the clusters. They arise because of the probabilistic basis for the selection of ants in the process of route construction. The next studied function is Rosenbrock function [29]. The equation of the n -dimensional Rosenbrock test

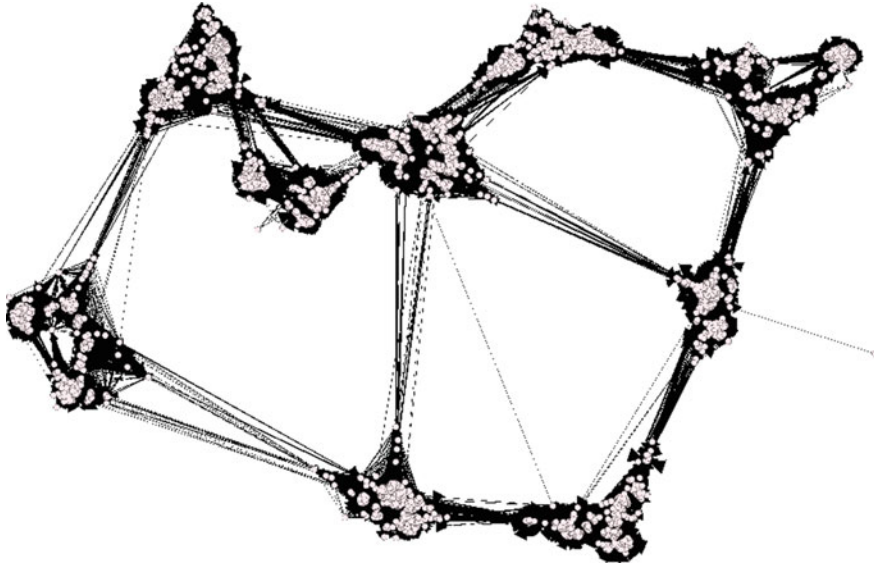


Fig. 2.21 Resulting graph

function is:

$$f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{n-1} \left[100 (x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right] \quad (2.7)$$

Its 2-dimensional view is represented in the Fig. 2.22.

Figure 2.23 shows the process of optimization of Rosenbrock function. The function was studied in the range $[-2.5; 2.5]$.

As it is known, Rosenbrock function has one global minimum, whose coordinates are $(x, y) = (1, 1)$, where $f(x, y) = 0$. The Fig. 2.23 shows that as a result of optimization a lot of resulting fragments has been received. This is due to the specific behavior of the function in these fragments and of course with the size the test space partition. Figure 2.24 shows the resulting graph.

The resulting graph consisting of clusters is obvious, but clusters are linked successively rather than all together like in Rastrigin function resulted in graph. Himmelblau function [30]. The equation of the n -dimensional Himmelblau test function is:

$$f_{\text{Himmelblau}}(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad (2.8)$$

Its 2-dimensional view is represented in the Fig. 2.25.

Figure 2.26 shows the process of optimization of Himmelblau function. The function was studied in the range $[-5; 5]$.

This function has 4 equal global optimums in points:

- $f(3.0, 2.0) = 0.0$

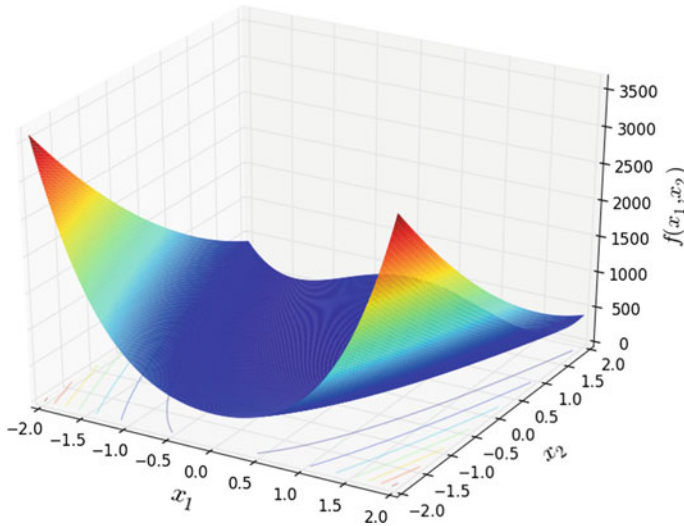


Fig. 2.22 Rosenbrock function chart

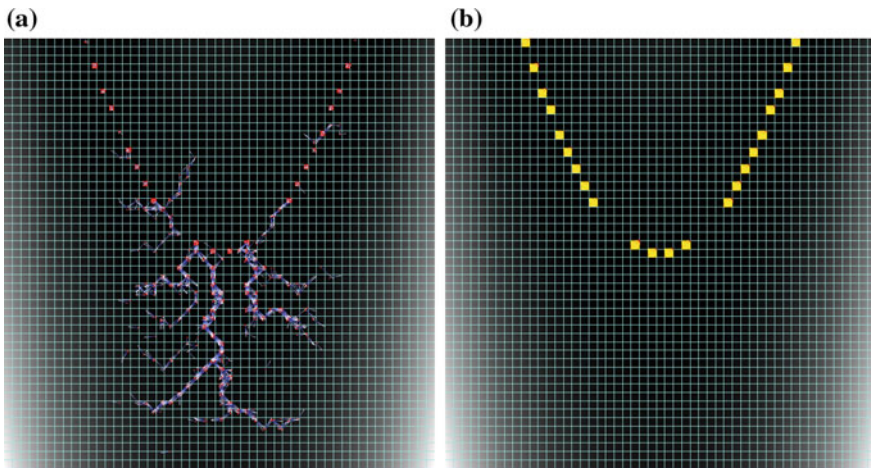


Fig. 2.23 Rosenbrock function optimization process: **a** middle step, **b** final result

- $f(-2.805118, 3.131312) = 0.0$
- $f(-3.779310, -3.283186) = 0.0$
- $f(3.584428, -1.848126) = 0.0$

All 4 extremes were localized. You can see them on Fig. 2.26b. Figure 2.27 shows the resulting graph.

Figure 2.27 shows all 4 localized extremes (4 clusters). Ursem01 function. The equation of the n -dimensional Ursem01 test function is:

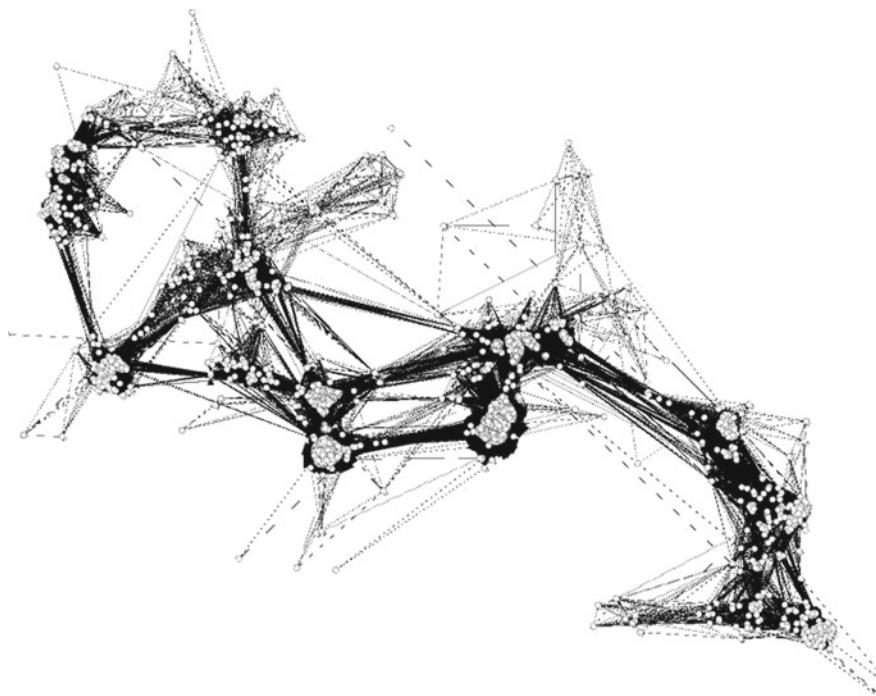


Fig. 2.24 Resulting graph

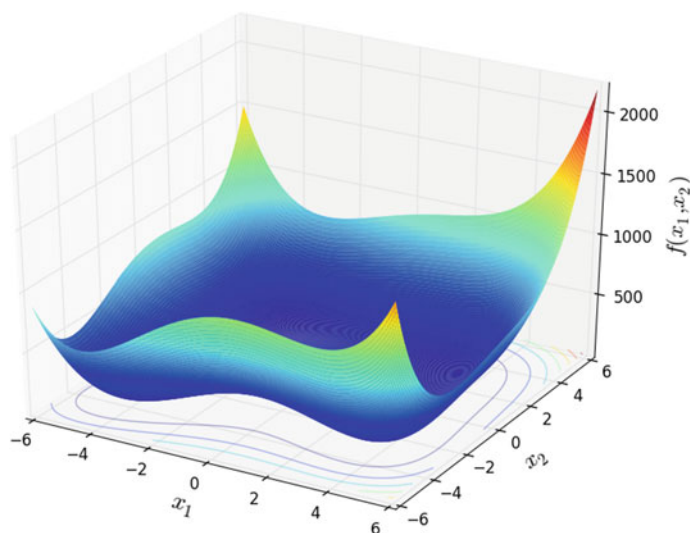


Fig. 2.25 Himmelblau function chart

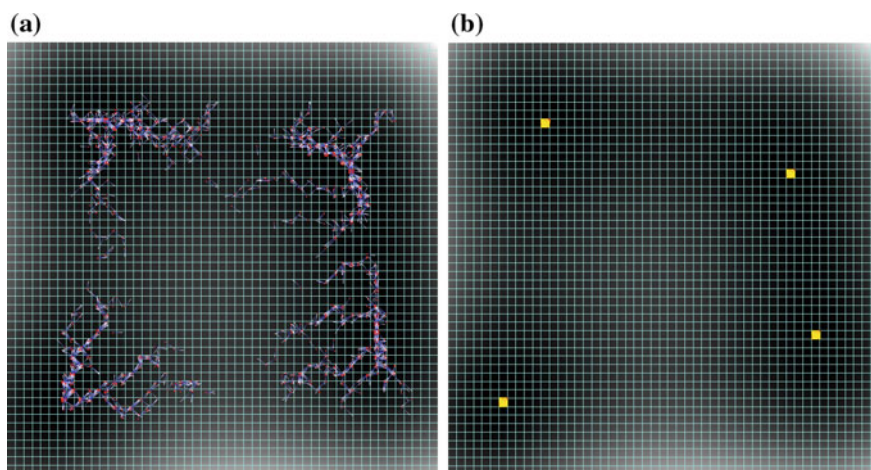


Fig. 2.26 Himmelblau function optimization process: **a** middle step, **b** final result

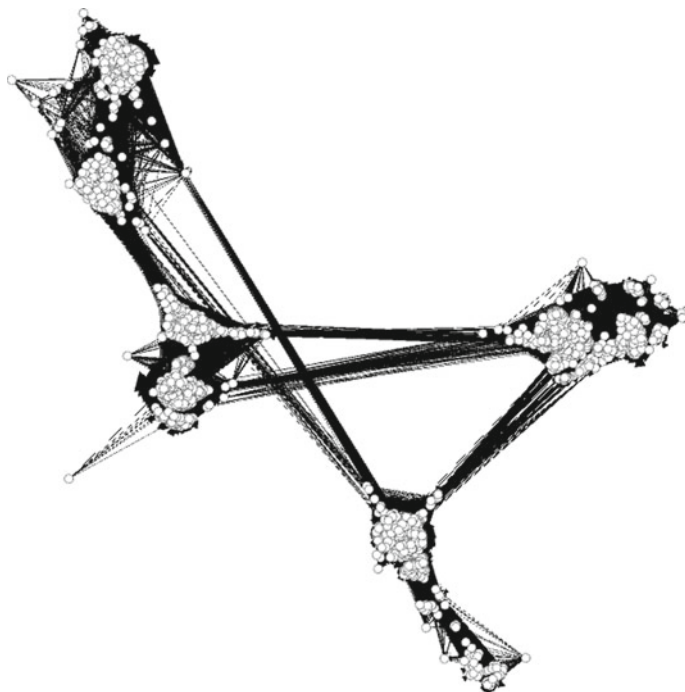


Fig. 2.27 Resulting graph

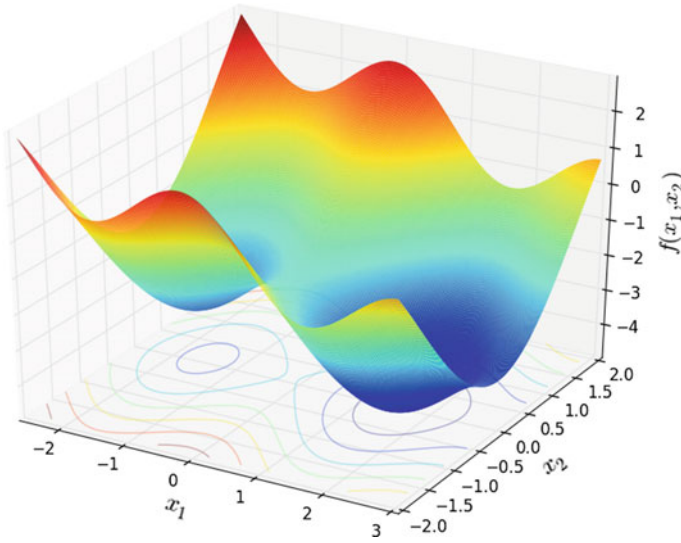


Fig. 2.28 Ursem01 function chart

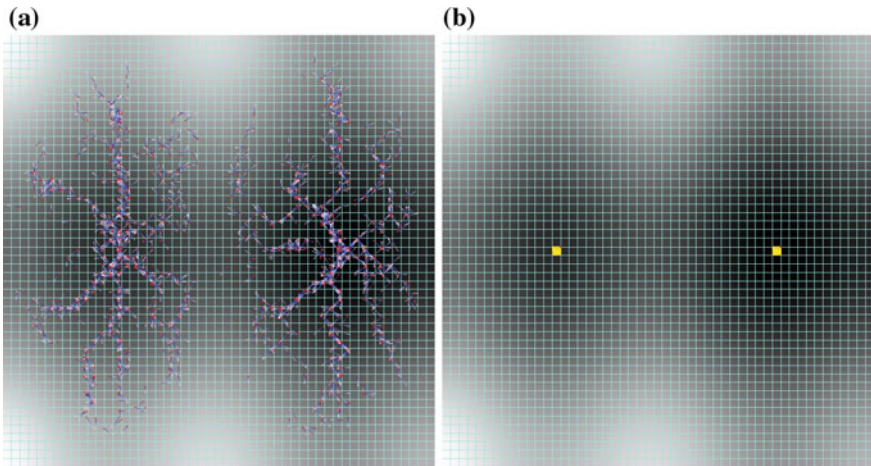


Fig. 2.29 Ursem01 function optimization process: **a** middle step, **b** final result

$$f_{Ursem01}(x) = -\sin(2x_1 - 0.5\pi) - 3\cos(x_2) - 0.5x_1 \quad (2.9)$$

Its 2-dimensional view is represented in the Fig. 2.28.

Figure 2.29 shows the process of optimization of a Userm01 function. The function was studied in the range $[-3; 3]$.

This function has one local and one global optimum in point with coordinates $[3, 2]$ where $f(3, 2) = 0.0$; Fig. 2.30 shows result graph.

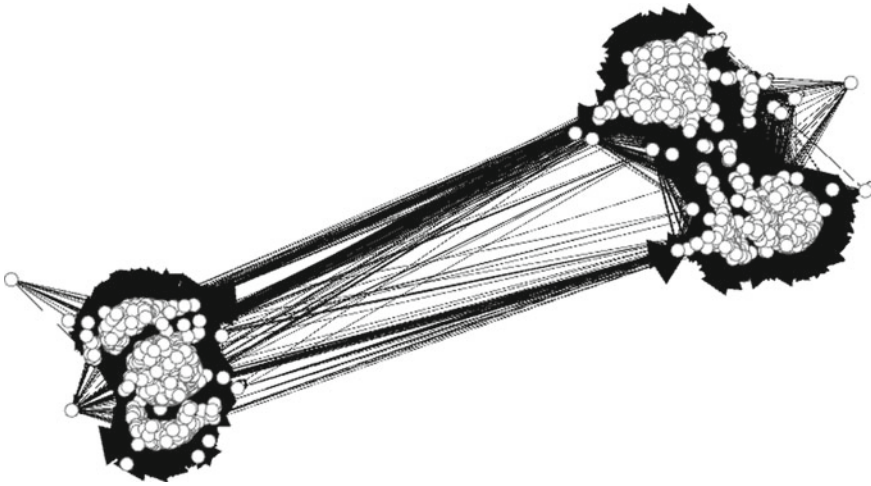


Fig. 2.30 Resulting graph

Figure 2.30 shows 2 clusters (2 extremes). One of them consists of more vertexes. It is characterized by the global extreme. Giunta function. The equation of the n -dimensional Giunta test function is:

$$f_{Giunta}(x) = 0.6 + \sum_{i=1}^n \left[\sin^2 \left(1 - \frac{16}{15}x_i \right) - \frac{1}{50} \sin \left(4 - \frac{64}{15}x_i \right) - \sin \left(1 - \frac{16}{15}x_i \right) \right] \quad (2.10)$$

Its 2-dimensional view is represented in the Fig. 2.31.

Figure 2.32 shows the process of optimization of Giunta function. The function was studied in the range $[-1; 1]$.

Giunta function has one global optimum $f(x) = 0.06447$ for $x = [0, 4673; 0, 4673]$. It is situated in top right part in Fig. 2.32b. There is a lot of localized fragments on the left and bottom borders of Fig. 2.32a and b. It happens because ants can move only to fragments with lower function value, so they cannot pass through the surface convexity to the global minimum and tend to move to borders. Figure 2.33 shows resulting graph.

Figure 2.33 shows one big cluster because Giunta function has only one extreme. Also, there are many vertexes situated far from the main cluster. It occurred because many ants accumulated on left and bottom borders. Schwefel26 function. The equation of the n -dimensional Schwefel26 test function is:

$$f_{Schwefel26}(x) = 418.9829n - \sum_{i=1}^n x_i \sin \left(\sqrt{|x_i|} \right) \quad (2.11)$$

Its 2-dimensional view is represented in the Fig. 2.34.

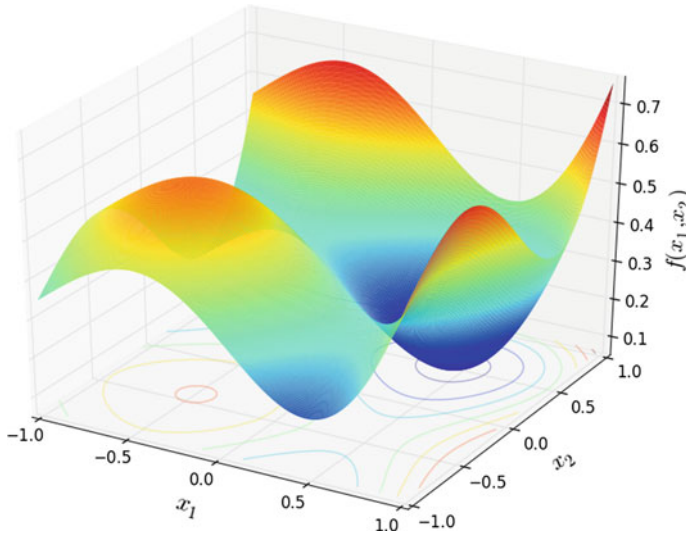


Fig. 2.31 Giunta function chart

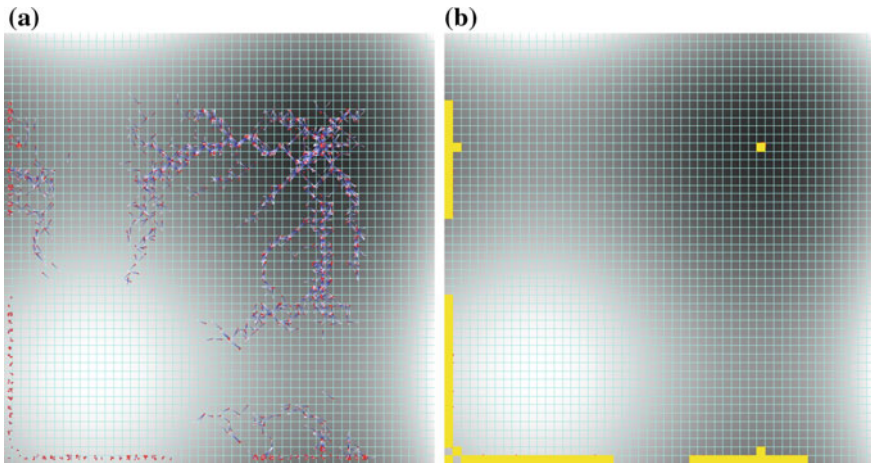


Fig. 2.32 Giunta function optimization process: **a** middle step, **b** final result

Figure 2.35 shows the process of optimization of a Schwefel26 function. The function was studied in the range $[-500; 500]$.

Schwefel26 function [31] has many extremes. Figure 2.35 proves it. All of them were localized. Figure 2.36 shows result graph.

As a result, authors got a graph with many clusters. Also, authors noted, that there are graphs which don't connect with the main graph. It means that in the process of localizing the exact extremes ants did not connect with other ants. It depends on the

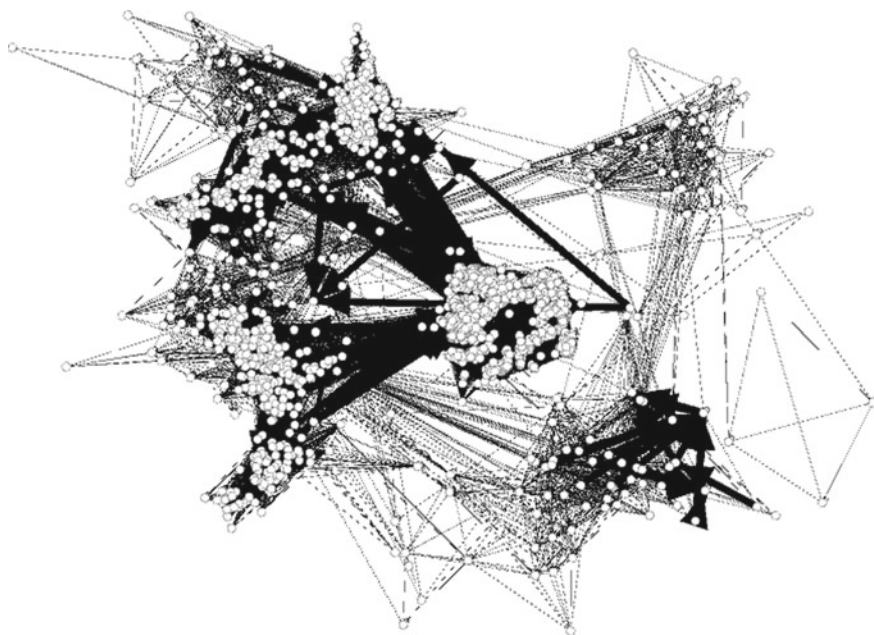


Fig. 2.33 Resulting graph

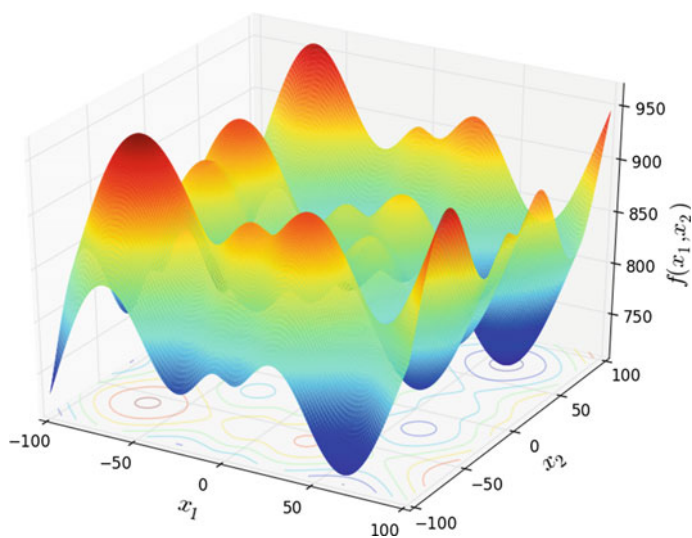


Fig. 2.34 Schwefel26 function chart

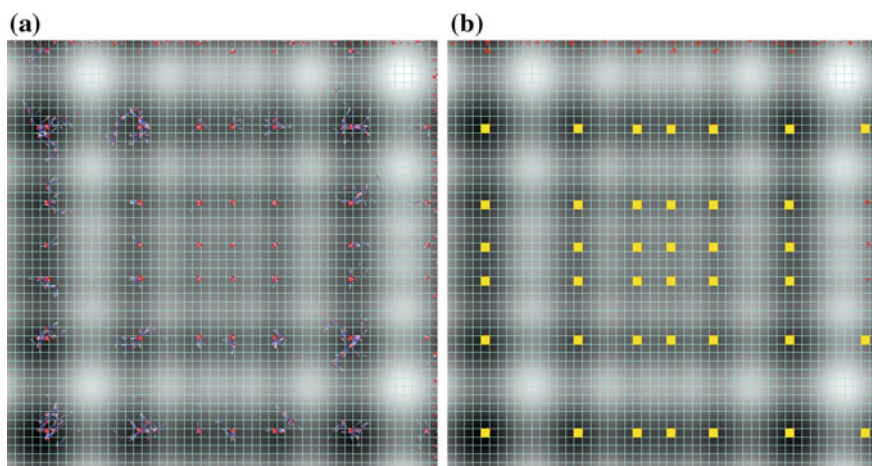


Fig. 2.35 Schwefel26 function optimization process: **a** middle step, **b** final result



Fig. 2.36 Resulting graph

Table 2.1 Detailed information about resulted graphs

Function	Avg. degree	Avg. weighted degree	Network diameter	Graph density	Modularity	Connected components	Avg. clustering coefficient	Avg. path length
Rastrigin	245.612	569.648	11	0.047	0.894	1	0.818	4.755
Rosenbrock	690.186	56343.303	10	0.135	0.624	1	0.874	3.461
Himmelblau	707.384	7523.683	7	0.136	0.820	1	0.885	3.125
Ursen01	800.477	154.119	5	0.270	0.878	1	0.878	2.211
Giunta	1044.556	0.618	9	0.207	0.933	1	0.780	2.511
Schwefel26	68.425	572.385	32	0.014	0.939	8	0.821	12.365

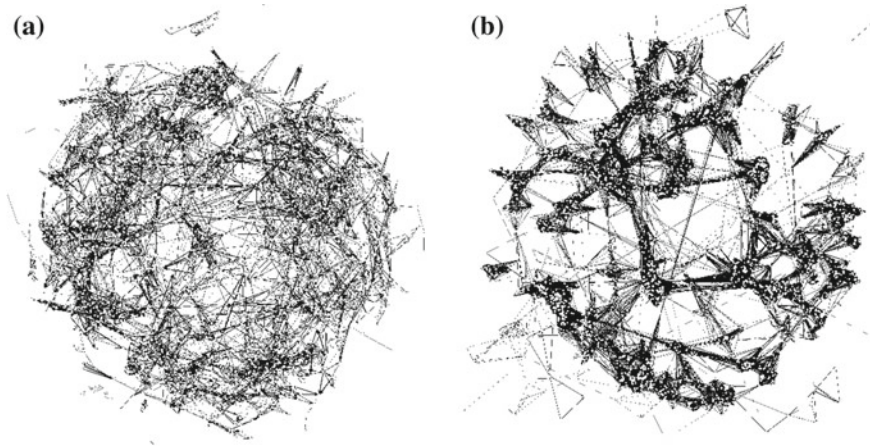


Fig. 2.37 Network formatting during the process of Schwefel function optimization: **a** 3rd iteration, **b** 5th iteration

size of the partition. More information [32] about the resulting graphs is shown in Table 2.1.

It is also possible to observe the dynamic change in the graph structure through function optimization. Figure 2.37 shows graph structure changes in the optimization process.

As you can see in the course of time clusters are formed. Figure 2.37a shows graph on 3rd iteration. Ants have just started to move, and haven't earned much experience from the colony. But after 5th iteration (see Fig. 2.37b) nodes form groups (clusters). All ants in the cluster are participating in the optimization of a certain extreme, and thus provide each other the greatest influence within the group.

2.4 Conclusion

The authors carried out series of experiments to analyze the behavior of ants. Special software tool for optimization of the test functions was developed using C# programming language and Microsoft Visual Studio. All experiments carried out on a PC with processor AMD Phenom II P960 with 6 GB of RAM. To analyze and to visualize data authors used free software tool Gephi. Following the results of experiments, we can conclude that there is a definite pattern in the behavior of ants and as a result the structure of the graph. There is the formation of clusters. It is also worth noting that there is a strong link between the ants within the group and a weak dependence on the ants of other clusters. The experiments allowed to illustrate the interaction of agents in the colony during the search for optimal solutions of test functions, such as Rastrigin, Rosenbrock, Himmelblau, Schwefel26, Giunta and Ursem01 functions.

Acknowledgements The following grants are acknowledged for the financial support provided to this research: Grant Agency of the Czech Republic - GACR P103/15/06700S, Grant of SGS No. SGS 2017/134, VSB-Technical University of Ostrava. The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

References

1. Zelinka, I.: SOMA self organizing migrating algorithm, Chap. 7. In: B.V. Babu, G. Onwubolu (eds.) *New Optimization Techniques in Engineering*, p. 33. Springer, Berlin (2004). ISBN 3-540-20167X
2. Zelinka, I., Davendra, D., Chadli, M., Senkerik, R., Dao, T.T., Skanderova, L.: Evolutionary dynamics and complex networks. In: Zelinka, I., Snasel, V., Ajith A. (eds.) *Handbook of Optimization*, p. 1100s. Springer, Germany (2012)
3. Zelinka, I., Davendra, D., Chadli, M., Senkerik, R., Dao, T.T., Skanderova, L.: Evolutionary dynamics and complex networks. In: Zelinka, I., Snasel, V., Ajith, A. (eds.) *Handbook of Optimization*, p. 1100s. Springer, Germany (2012)
4. Price, K.: An introduction to differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimization*, p. 79108. McGraw-Hill, London (1999)
5. Donald, D., Zelinka, I.: Self-organizing migrating algorithm. *New Optimization Techniques in Engineering* (2016)
6. Metlicka, M., Davendra, D.: Chaos-driven discrete artificial bee colony. *IEEE Congress on Evolutionary Computation*, pp. 2947–2954 (2014)
7. Davendra, D., Zelinka, I., Metlicka, M., Senkerik, R., Pluhacek, M.: Complex network analysis of differential evolution algorithm applied to flowshop with no-wait problem. In: *IEEE Symposium on Differential Evolution*, 2014, Orlando, FL, USA, pp. 65–72, 9–12 December 2014
8. Davendra, D., Metlicka, M.: Ensemble centralities based adaptive artificial bee algorithm. In: *IEEE Congress on Evolutionary Computation* (2015)
9. Zelinka, I.: Evolutionary algorithms as a complex dynamical systems. In: *Tutorial at IEEE Congress on Evolutionary Computation 2015*, Sendai (2015)
10. Zelinka, I.: On mutual relations amongst evolutionary algorithm dynamics, its hidden complex network structures. In: Meghanathan, N. (ed.) *Advanced Methods for Complex Network Analysis*, IGI, An Overview and Recent Advances (2015)
11. Skanderova L., Zelinka I.: Differential evolution dynamic analysis by the complex networks. In: Meghanathan, N. (ed.) *Advanced Methods for Complex Network Analysis*, IGI (2015)
12. Meyn, S.: *Control Techniques for Complex Networks*. Cambridge University Press, Cambridge (2007)
13. van Steen, M.: *Graph Theory and Complex Networks: An Introduction*. Maarten van Steen, Amsterdam (2010)
14. Chen, G., Wang, X.: *Xiang Li Fundamentals of Complex Networks: Models, Structures and Dynamics* (2015)
15. Zelinka, I., Davendra, D., Senkerik, R., Jasek, R.: Do evolutionary algorithm dynamics create complex network structures? *Complex Syst.* **2**, 0891–2513, **20**, 127–140 (2011)
16. Zelinka, I., Davendra, D., Snasel, V., Jasek, R., Senkerik, R., Oplatkova, Z.: Preliminary Investigation on Relations Between Complex Networks and Evolutionary Algorithms Dynamics, CISIM, Poland (2010)
17. Zelinka, I., Davendra, D., Chadli, M., Senkerik, R., Dao, T.T. Skanderova, L.: Evolutionary dynamics and complex networks. In: *Handbook of Optimization*. Springer Series on Intelligent Systems (2012)

18. Michal, P., Jakub, J., Roman, S., Ivan, Z., Donald, D.: 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. Springer International Publishing, Berlin (2016)
19. Michal, P., Roman, S., Jakub, J., Adam, V., Ivan, Z.: Study on swarm dynamics converted into complex network. In: Proceedings of the 30th European Conference on Modelling and Simulation, ECMS 2016, European Council for Modelling and Simulation (ECMS) (2016)
20. Michal, P., Roman, S., Adam, V., Jakub, J., Donald, D.: Complex network analysis in PSO as an fitness landscape classifier. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 3332–3337. IEEE (2016)
21. Michal, P., Roman, S., Adam, V., Ivan, Z.: Creating complex networks using multi-swarm PSO. In: 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 180–185. IEEE (2016)
22. Zelinka, I.: Controlling complexity. In: Sanayei, A., Zelinka, I., Rossler, O.E. (eds.) ISCS 2013: Interdisciplinary Symposium on Complex Systems, Emergence, Complexity and Computation, vol. 8. Springer, Berlin (2014)
23. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**(1), 53–66 (1997)
24. Liu, X., Fu, H.: An effective clustering algorithm with ant colony. *J. Comput.* **5**(4), 598–605 (2010)
25. Toksari, M.D.: Ant colony optimization for finding the global minimum. *Appl. Math. Comput.* **176**, 308316 (2006)
26. Neydorf, R.A., Yarakhmedov, O.T.: Development, optimization and analysis of parameters of classic ant colony algorithm in solving travelling salesman problem on graph. *Sci. Technol. Prod.* **3**(2), 18–22 (2015)
27. Hu, Y.: Efficient High-Quality Force-Directed Graph Drawing. *Math. J.* **10**(1), 37–71 (2006). Wolfram, Media
28. Rastrigin, L.A.: Systems of Extremal Control. Nauka, Moscow (1974). (in Russian)
29. Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. *Comput. J.* **3**, 175184 (1960)
30. Himmelblau, D.: Applied Nonlinear Programming. McGraw-Hill, New York (1972)
31. Laguna, M., Marti, R.: Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions (2002)
32. Newman, M.: Networks: An Introduction. Oxford University Press, Oxford (2010)

Evolutionary Algorithms, Swarm Dynamics and Complex
Networks

Methodology, Perspectives and Implementation

Zelinka, I.; Chen, G. (Eds.)

2018, XXII, 312 p. 194 illus., 155 illus. in color.,

Hardcover

ISBN: 978-3-662-55661-0