Comparative Analysis of Heuristic Algorithms for Solving Multiextremal Problems

Rudolf Neydorf, Ivan Chernogorov, Victor Polyakh
Orkhan Yarakhmedov, Yulia Goncharova
Department of Software Computer Technology and Automated Systems and Department of Scientific-Technical Translation and Professional Communication
Don State Technical University
Rostov-on-Don, Russia

Email: ran_pro@mail.ru, hintaivr@gmail.com, silvervpolyah@gmail.com, orhashka@gmail.com, jl.goncharova@gmail.com

Dean Vucinic
Vesalius College
Vrije Universiteit Brussel
Brussels, Belgium

Email: dean.vucinic@vub.ac.be

***Abstract*—In this paper, 3 of the most popular search optimization algorithms are applied to study the multi-extremal problems, which are more extensive and complex than the single-extremal problems. This study has shown that only the heuristic algorithms can provide an effective solution to solve the multiextremal problems. Among the large group of available algorithms, the 3 methods have demonstrated the best performance, which are: (1) particles swarming modelling method, (2) evolutionary-genetic extrema selection and (3) search technique based on the ant colony method. The previous comparison study, where these approaches have been applied to an overall test environment with the multiextremal Rastrigin functions, has shown already their suitability to solve multiextremal problems. In addition, they are characterized with superior performance properties. Nevertheless, each of the selected heuristic algorithms has demonstrated its own specific search features that allow the detection and identification of both global and local extremes. In this paper, the investigated algorithms have been validated on a larger test functions environment with different types of extremes. The particular attention was given to analyse their individual methods when solving the data-clustering problem. The main conclusion is that each of these methods can find the extremes by satisfying any desired precision and have acceptable performance, when applied to the variety of practical problems.**

Keywords—searching optimization; multi-extremes; genetic algorithm; swarm algorithm; ant algorithm.

I. INTRODUCTION

For the current state of the theory of optimization is quite common that most of the known methods are designed to find only the global optima. Many of these methods are highly effective [1][2][3][4]. At the same time, the scope of the optimization methods, and related application areas are continuously expanding, as being part of the most advanced areas in science and technology. In addition, many social and economic projects, military and other applications are almost always faced to the formulation of optimization problems for which more precise solutions are needed.

Many modern practical optimization problems are inherently complicated by counterpoint criterion requirements of the involved optimized object. The expected result - the global optimum - for the selected criteria is not always the best solution to consider, because it incorporate many additional criteria and restrictions. It is well known that such situations arise in the design of complex technological systems when solving transportation and logistics problems among many others. In addition, many objects in their technical and informational nature are prone to multi-extreme property. In particular, these objects and the discrete nature of their respective systems have significant multi-extreme property (ME) [5][6][7][8][9][10] [11][12].

A distinctive approach for solving such problems requires iterative steps to evaluate a large number of options in order to shape and find the solutions. The result of this process is that the developers are forced to apply search engine optimization (SO) [2][3][4].

In the second half of the last century and at the beginning of this century, the theoretical research and the practical application of their results have shown that it is inappropriate to find such methods in the class of so-called deterministic methods, as many attempts in following such approach have resulted to be ineffective. The reason is that these techniques are too sensitive to non-smoothness and other characteristics that are encountered when having continuous dependency, while as well-known, the problems related to the discrete programming lead to the application of the NP-complete algorithms.

Therefore, to solve many practical optimization problems, especially problems of ME, it is appropriate to apply the so-called heuristics methods. These methods, according to the authors, are the most promising for solving the discussed ME problems [6][7][8][9][10][11][12].

A. Formulation of the problem

As mentioned above, the motivation is to apply the most common heuristic SO methods to the environment having more typical, universal and complex ME problem, which has to be solved. The performed research revealed the possibility of finding some or all the extremes by applying

the selected methods. For this qualitative evaluation, it is necessary to numerically assess the accuracy of the found extremes values, as well as, the accuracy of their coordinates. Therefore, in the first stage of this research, we suggest the ME test function that might provide a common evaluation environment for validating the selected methods, when solving the proposed ME tasks. In the second stage of this research, the exact heuristic approaches are chosen, in order to determine both the well-known methods of solving ME tasks and their implementation algorithms.

B. Choosing multiextremal test function with a preliminary analysis of its properties

The most common and effective test functions for developing and analysing the SO methods are the Rosenbrock, Himmelblau and Rastrigin functions. The Rastrigin function (RF) is the most widely applied ME function between all of them. This universal function is not convex, as already shown in 1974 by Rastrigin [13]. The equation of N function arguments is:

$$f(x) = A \cdot n + \sum_{i=1}^{n} \left[x_i^2 - A \cdot \cos(2 \cdot \pi \cdot x_i) \right], \qquad (1)$$

where: $x=(x_1,\ldots,x_n)^T$ – vector; A=10.

The global minimum of this function is at the point (0,0)=0. It is difficult to find a local minimum of this function, because it has many local minimums. The isolation and evaluation of extremes is a complex task.

In Section II, the 3 most popular approaches of finding the set of extreme problems are discussed for the 2-dimensional Rastrigin function. Section III describes the related work. In Section IV, the conclusion of the conducted research is given.

II. SELECTING A GROUP OF HEURISTIC METHODS

In this paper, the authors established the 3 most relevant tasks, which are common in practice, when solving various search optimization tasks.

A. RF using swarming particles method

The essence and reasons in using the method of swarming particles (MSP) in SO tasks is well known [14][15][16][17][18]. The classic MSP algorithm simulates the real behaviour patterns of insects, birds, fishes, many protozoa, etc. However, ME objects require to know some specific properties of this algorithm.

The authors of [19][20][21] and other members of R. Neudorf team [8][9][10][11][12] have significantly reworked the canonical MSP algorithm. In particular, a new modified version of this algorithm was developed for solving the ME tasks, which is based on a model based on the mechanical principles of the moving particle, and complemented by the mechanisms borrowed from the biological laws, as well as, the method of adaptation mechanisms, being property of the ME task.

The Mechanical Movement Model (MMM) of particles [21] in MSP was significantly modified and refined:

$$X_{ti} = X_{(t-\Delta t)i} + \vec{V}_{(t-\Delta t)i} \cdot \Delta t\,, \qquad (2)$$

$$\vec{V}_{ti} = \vec{V}_{(t-\Delta t)i} + \vec{A}_{(t-\Delta t)i} \cdot \Delta t\,, \qquad (3)$$

$$\vec{A}_i = \vec{F}_{pi} + \vec{F}_{tri}\,, \qquad (4)$$

where: $X_{(t-\Delta t)i}$ – i-th particle previous position; X_{ti} – i-th particle current position; V_{ti} – i-th particle velocity at the current time; $V_{(t-\Delta t)i}$ – i-th particle current velocity; $A_{(t-\Delta t)i}$ – particle previous acceleration in previous time; Δt – integration interval; F_{pi} – acceleration caused by the particles biologically action attractive forces; F_{tri} – slowing under the action of friction forces.

F_{pi} – acceleration caused by the particles biologically action attractive forces includes 3 sub-attractions:

$$\vec{F}_{pi} = \vec{F}_{pi}^{G} + \vec{F}_{pi}^{L} + \vec{F}_{pi}^{C}\,, \qquad (5)$$

where: F^G_{pi} - particle attraction to global extreme; F^L_{pi} - particle attraction to the local extreme of particle (the best finding position by particle during its existence); F^C_{pi} - particle attraction to the nearest cluster.

The sub-attraction in the described algorithm is based on an analogue of the law of gravitational attraction:

$$\vec{F}_{pi}^{Q} = \frac{\vec{G}^{Q} m_i m_e}{r^2}\,, \qquad (6)$$

where: $Q \in \{G, L, C\}$, G^Q – the proportionality coefficient (gravity prototype); m_i - the desire measure of i^{th} particle to the selected best particle with bio-similar of m_e mass for the attractive particle (as a "bee flies to the womb"); r - the distance between the current position of the particle and extrema.

In order to eliminate the errors (occurring at r=0 and $r \rightarrow 0^+$) the following changes are introduced:

- when the particle is updating the global, local or cluster extreme, it loses one or more sub-attraction, because it is currently located in the best position (global, local or cluster) and thus continues the movement at the expense of the remaining sub-attractions or inertia;
- when the particle is at the point of the current global, local or cluster extreme (r=0). The limitation is naturally set in MM by formula (6):

$$\vec{F}_{pi}^{Q} = \frac{\vec{G}^{Q} m_i m_e}{r^2 + \varepsilon}\,,$$

where ε - setup option, limiting the maximum acceleration, delaying the passage of the actual (and finding) particles from the centre of gravity;

- when the particle moves too close to the global, local or cluster extreme ($r \to 0^+$). The particle gets a great acceleration that causes an increase in resistance of the medium (F_{tri}) and limits the maximum acceleration/speed.

To improve the searching properties, the stochastic blur parameter was introduced:

$$\lambda^{\xi}(\varepsilon) = \lambda \cdot (1 + 2 \cdot \varepsilon(rnd(1) - 0.5)), \quad (7)$$

where: $\lambda^{\xi}(\varepsilon)$ – fluctuating parameter value at each iteration; ε – distorted relative deviation parameter from nominal value; $rnd(1)$ – random number in the range [0, 1].

MSP contains the reflect mechanism. The particles reflect within the boundaries ranges of the selected function. This increases the area under investigation when particles try to "fly" over the treated area.

Initially, the authors had decided to implement a dynamic clustering mechanism, which would allow particles to localized extremes, to further improve the search results, by swarming around the found local and global extremes. However, after preliminary research, authors decided to implement the clustering mechanism that is a combination of the 2 concepts - kinematic and dynamic. The kinematic concept is expressed at each iteration where the positions of all particles together with the previously created clusters points undergo the clustering ("A quasi-equivalence" algorithm [22][23]). This mechanism allows selecting the area of all found global and local extremes (the number of localized extremes may not exceed the number of particles * number of carried out iterations), by selected criteria.

"A Quasi Equivalence" clustering algorithm does not require resulting number of clusters. It can be described by the following equations, which is the matrix of normal similarity measures:

$$\mu_{x_q}(x_i) = 1 - \frac{d(x_q, x_i)}{\max\limits_{k \in [1,Q]} (d(x_q, x_k))}, \quad (8)$$

where: x – is the plurality of elements; Q – is a number of elements in plurality; $(q, i) = \{1, Q\}$; $d(x,y)$ – is a clustering criterion (like Euclidean distance between points or etc.).

The relative similarity measures are defined with:

$$\xi_{x_q}(x_i, x_j) = 1 - \left| \mu_{x_q}(x_i) - \mu_{x_q}(x_j) \right|, \quad (9)$$

where: $(i, j, q) = \{1, Q\}$.

The matrix of similarity measures of the elements plurality:

$$\xi(a,b) = T(\xi_{x_1}(a,b), \ldots, \xi_{x_Q}(a,b)) = \\ = \min_{i=1,Q} \xi_{x_i}(a,b), \quad (10)$$

where: $a, b \in X$.

The result matrix:

$$R_{\xi}^{q} = R_{\xi}^{q-1} \circ R_{\xi}, \quad (11)$$

where: S=max, T=min.

The values in the R matrix show whether the pair of points belongs to the R relation, called "quasi-equivalence levels" - a. The choice of a particular level divides the plurality into equivalence classes, which correspond to the separate clusters. Fig. 1 demonstrates the flowchart of "A Quasi Equivalence" clustering.

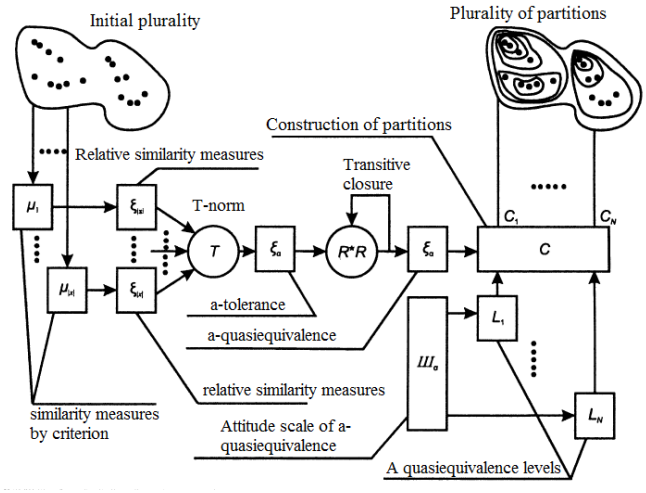


Figure 1. "A quasiequivalence" algorithm flow-chart

The ME MSP modification requires the "A Quasi Equivalence" clustering based on the Euclidean distance between the allocated extremes criteria. After this action, all the points in the considered clusters are deleted, except the extreme point, which allows to dropout the sub-local values.

The dynamic concept consists of the following steps: after the kinematic clustering particles appear with the "attractive force" to the extreme areas of the whole swarm, not only the global extreme, found as the best position for the particle. In this paper, the authors have chosen a strategy of particles attraction to be the centre of the nearby cluster, as it allows them to react instantly to changing situations (the emergence of new areas to find extreme).

To test and debug MSP, the authors have developed the software tool «MMSP» (implemented by I. Chernogorov), which has enhanced functionality. The tool is implemented in C#. Fig. 2 shows the part of MMSP interface (without sub-area, which visualized the selected function, particles and created clusters. The Canvas and Helix library were used to display it. Authors used different libraries, because the 3D scene heavy loads the PC, which is not intended for huge experiments. Fig. 3(a) and 3(b) display the visualization of Guinta function, position and velocity vectors of the particles, and created clusters for 2D and 3D scenes), highlighting the diverse areas to display information.

MMSP workspace is divided into the following sub-areas:

- orange rectangle – MMSP sub-area, is responsible for the initialization of particles, the iteration (in the "step by step" and "automatic" mode) and restarting the computation;
- green rectangle – MMSP sub-area, in which the user selects the desired test function and variable ranges;
- blue rectangle - MMSP sub-area, in which the user sets up the 2D or 3D display mode by selecting the function and/or particles and/or created clusters;
- purple rectangle – MMSP sub-area, is responsible for the customization of MSP parameters;
- pink rectangle – MMSP sub-area, showing the MSP results at current moment: the global extreme computing time of initialization, iteration and clustering, number of the current iteration and the number of test function calls.

The testing modifications effectiveness was carried out for RF in coordinate range $(x,y) \in [-1.5, 1.5]$. In this area, RF has 9 local minimums, including one global. Fig. 4(a), 4(b) and 4(c) show extreme areas localization process with kinematic-dynamic clustering and the creation of the corresponding clusters.

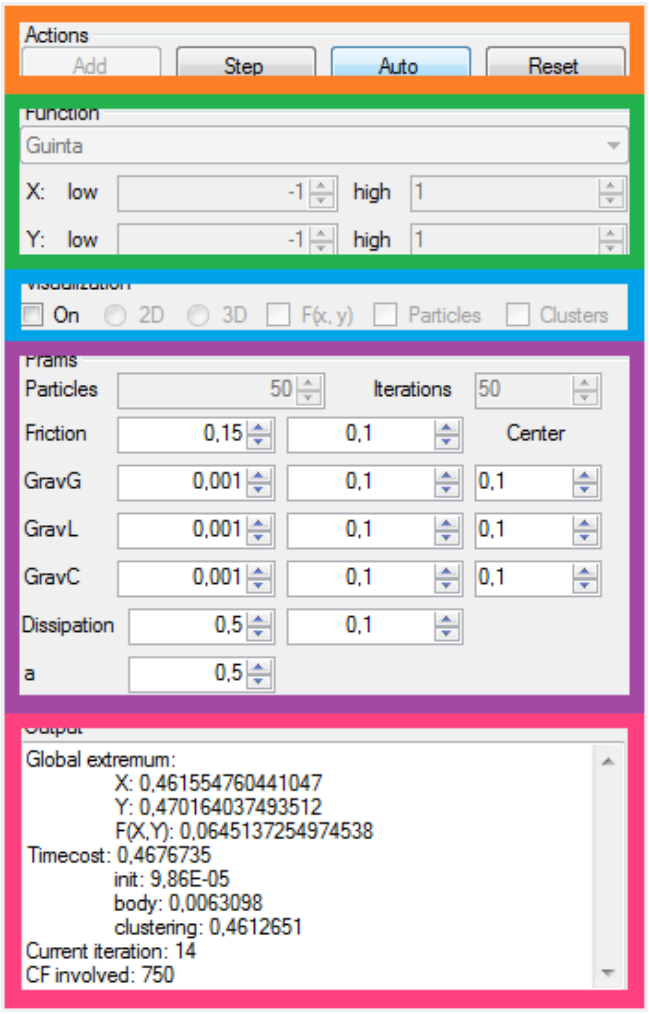
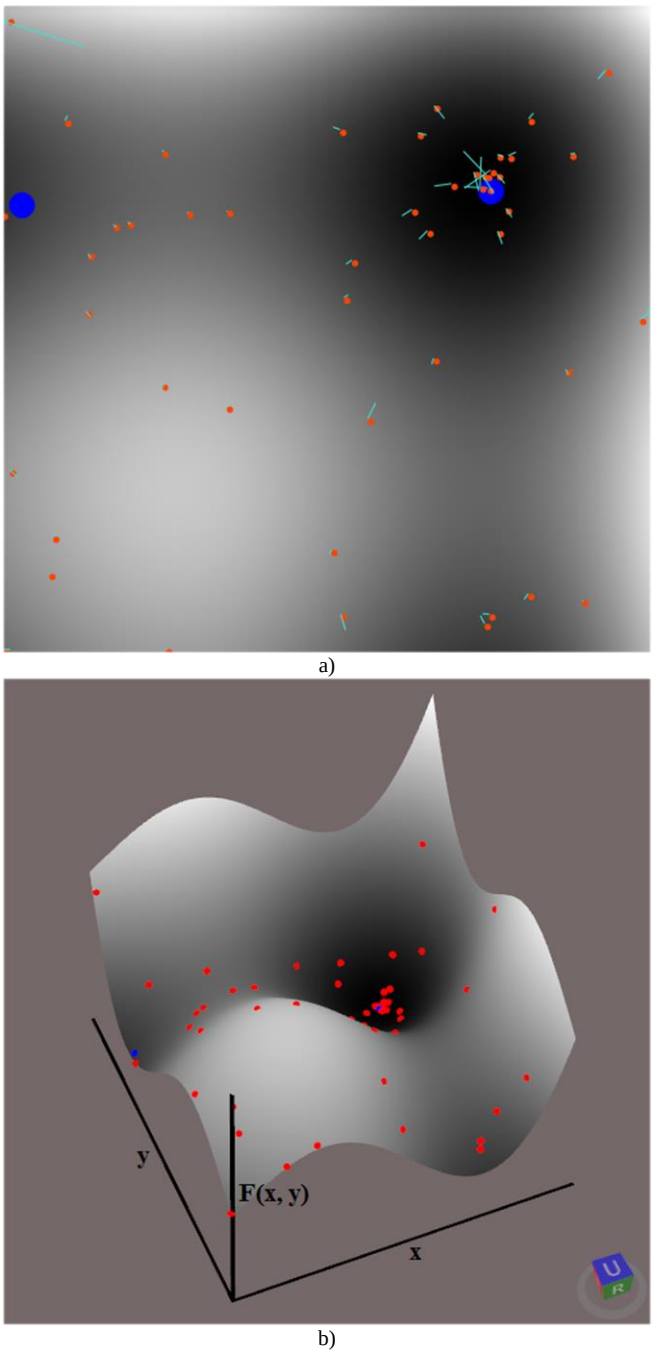


Figure 2. Part of MMSP interface.

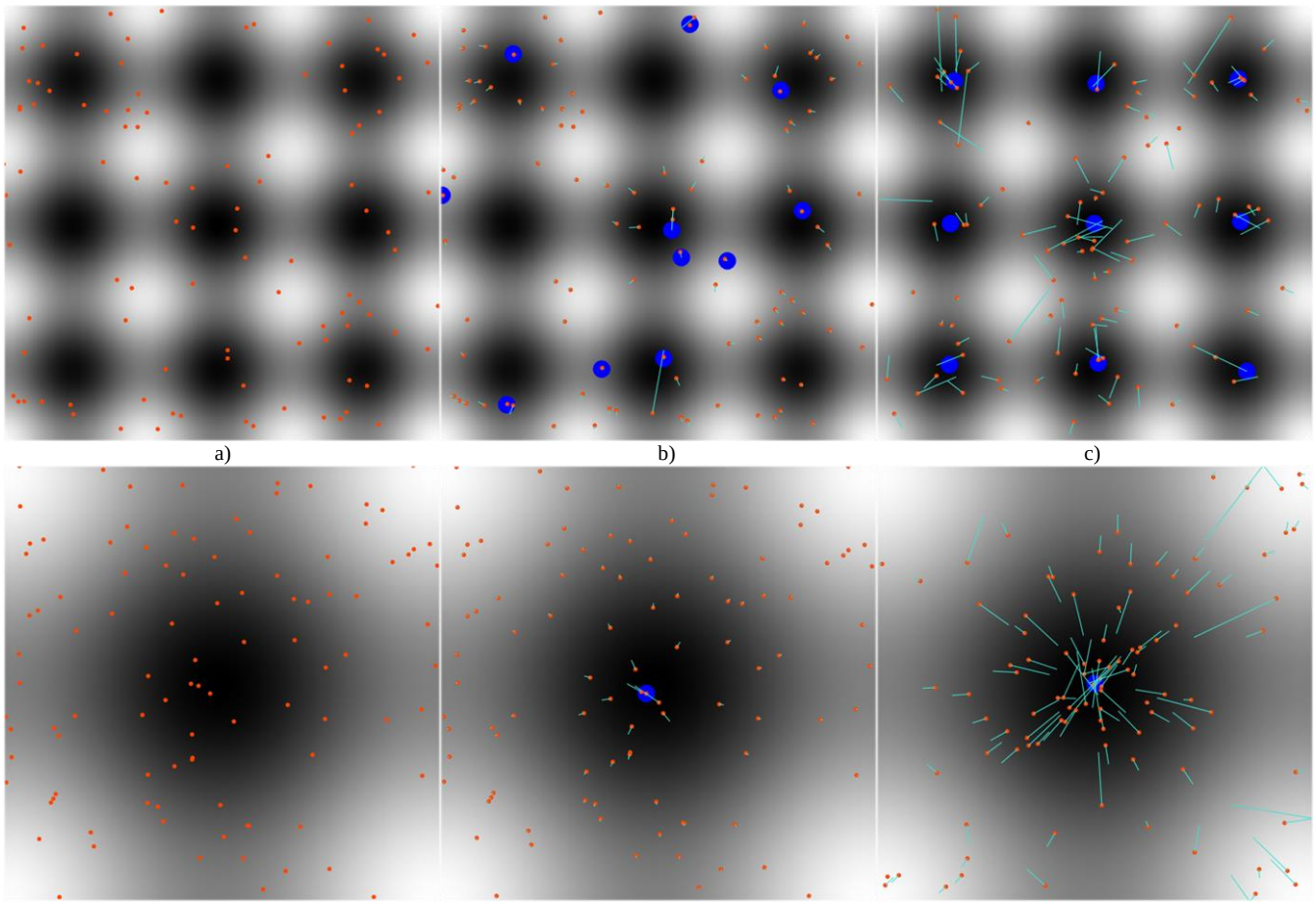


a)

b)

Figure 3. Visualization of the function, particles and clusters in MMSP on (a – 2D scene; b – 3D scene).

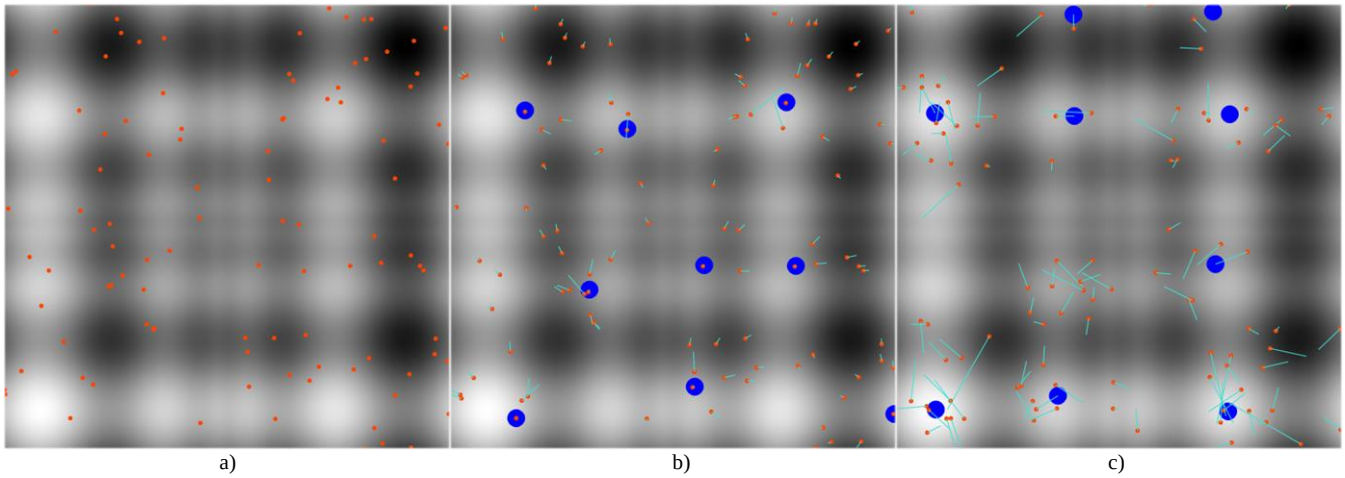
Fig. 4(a), 4(b) and 4(c) show that the particles are initially attracted to the resulting cluster, which is located in the global extreme area. This is due to the overall prevalence of the global attraction power over the local forces of attraction. Some peripheral particles might find the local extremes, which are attracted to them, and gathered in clusters. In strict clusters areas, the ME MSP algorithm (in case of having less isolated and significant extremes) is repeated.



a) b) c)

d) e) f)

Figure 4. Extremal RF areas localization of (a – the initialization, b – the 1st iteration, c – the 50th iteration). RF local identification of global extreme of (d – the initialization, e – the 1st iteration, f – the 50th iteration)



a) b) c)

Figure 5. Extremal Schwefel_26 function areas localization of (a – the initialization, b – the 1st iteration, c – the 50th iteration).

This process is iteratively repeated until the desired accuracy of the local and global extreme parameters is achieved. Within the limited time for fulfilling the algorithm of each cluster, a quite stable dynamic equilibrium of particles is set. The calculations, for the modelling activity, make obvious that the average number of the particles is correlated with the value of the extreme. The degree of correlation depends on the ME MSP algorithm settings.

In order to improve the accuracy of any extreme parameters estimation, the repetition of ME MSP algorithm

is applied to the contracted areas of the defined clusters. This process can be iteratively repeated until the desired accuracy is achieved, by taking into account all the local and global extremes.

The examples in Fig. 4(d), 4(e) and 4(f) demonstrate the fragments of the iterative identification of the global extreme, which is located at the point [0, 0]. TABLE I shows the results obtained by the localization and additional identification in all areas. The table presents the coordinates $x=x_1$ and $y=x_2$, and the RF values obtained by applying the equation (2). The increase of number of iterations (and the search time) improves the estimated accuracy. Searching tasks carried out on the PC with AMD Phenom II P960 processor and 6Gb of RAM. At the same time, to achieve the described accuracy (localization of all extremes areas and additional identification of 9 areas) took ~ 32 seconds. Thus, we can conclude that ME MSP is an effective tool for solving the ME tasks.

TABLE I. RESULTS OF THE EXPERIMENT

Standard			**Extremal evaluation item**		
x	*y*	*f (x, y)*	*Coordinates*		*Value*
			x	*Y*	*f (x, y)*
-1	1	2	-1.00007	1.0001	2.000382825
-1	0	1	-1.0001	-0.0004	1.000292529
-1	-1	2	-1.00001	-1.0003	2.000595233
0	1	1	-0.0001	1.00009	1.000177161
0	0	0	-0.0005	0.0002	0.000049304
0	-1	1	-0.0006	-1.0003	1.000659723
1	1	2	1.0005	1.0008	2.002827059
1	0	1	1.0003	-0.002	1.001544741
1	-1	2	1.0003	-1.0004	2.001314045

Additional testing modifications effectiveness was carried out for more asymmetric Schwefel_26 test function [24] in coordinate range $(x,y) \in [-250, 250]$. The equation of N function argument is:

$$f(x) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|}) . \qquad (12)$$

Search format was changed: extremes – the highest values. Fig. 5(a), 5(b) and 5(c) show MSP work on different stages. To select a larger number of local extremes it is needed to optimize the *a* clustering parameter.

The modification of the kinematic-dynamic clustering mechanism allows reducing the time and increasing the search accuracy. In subsequent papers the authors decided to carry out modification of the clustering mechanism, in the direction of a dynamic paradigm, to give the particles more resemblance to a real prototype, expecting to improve the search results and to reduce the computing clustering time.

B. Features of the evolutionary-genetic algorithm.

In solving the search engine optimization problems [25][26][27], one of the most popular, proven and, therefore, demanded tools is the evolutionary genetic algorithm (EGA). The structure of classical EGA, its respective components, and their processing operators are well known. However, depending on the objective applications, EGA can be characterized by considerable structural parametric features. In particular, the use of EGA for solving ME problems [28][29][30][31], as shown by studies [28][29][30], requires the addition of classical EGA, which application is based on the assessment and extremes selection tools. The evaluation and selection are necessary to identify the type of the extreme (maximum, minimum), and for measuring their size. Furthermore, it is necessary to determine the position of extreme in the factor space, i.e., coordinates.

For the Clustering Algorithm, we develop an approach for the selection of extremes, based on one-sample Student t-test criteria [30][31][32]. The proposed approach involves the implementation of 2 sequential stages: 1 - generation and 2 - evolutionary selection of populations by EGA and subsequent clustering to receive its finishing generations results - the fittest. The obtained results, in the form of quantitative assessments, identified the extremes distributed over the coordinate groups, by checking them in respect to the 0-hypothesis.

The clustering algorithm, implemented in this approach, is a logical comparison of the obtained vectors $\vec{v}_i$ - results of evolutionary individual's selection with the average $\vec{v}_0$ vector for each cluster sample and considers an expectation estimate to find the real extreme. The application of the theoretical positions of 0-hypothesis by using one sample Student t-test takes a decision about the inclusion or non-inclusion of the individual within a cluster sampling. As a result, the clusters are formed from individuals, which structure corresponds to the known necessary and sufficient conditions, for the existence of a local extreme.

Conceptually, these conditions are set to have in each cluster the best individual and the best estimate (estimate of local extreme $\vec{v}_e$) for the sample, and the remaining individuals are forming the extreme neighbourhood. For the neighbouring individuals the sufficient conditions for the extreme $\vec{v}_e$ is to fulfil one of the 2 conditions:

- if $\vec{v}_e$ - maximum, then

$$\forall i \neq e \rightarrow \varphi(\vec{v}_i) < \varphi(\vec{v}_e) , \qquad (13)$$

- if $\vec{v}_e$ - minimum, then

$$\forall i \neq e \rightarrow \varphi(\vec{v}_i) > \varphi(\vec{v}_e) , \qquad (14)$$

where $\varphi(\cdot)$- function for which extremes are sought.

For implementation of the algorithm, in order to be able to estimate one sample, 0-hypothesis requires testing of the toiletries in each formed EGA vector of arguments $\vec{v}$ with each cluster in the finish population.

$$V_k = \{\vec{v}_{ki} \mid i \in [1, n_k]\} . \qquad (15)$$

However, the form of multi-dimensional vector based argument makes it impossible to be directly applied to one

sample Student t-test, formulated for the treatment of the scalar arrays.

In connection with this algorithm, the transformation of the vector quantities is implemented for their scalar evaluation. The main vector estimates of cluster V_k are averaged over a cluster sample vector $\vec{v}_0$ and the vector of local extreme evaluating $\vec{v}_e$. The main scalar evaluations of cluster V_k are metric estimation of the discrepancy vectors (the distance between the points in the factor space). A measure of the audited individual proximity between the coordinates of the vector $\vec{v}$ and the cluster is a unit vector of its deviation from $\vec{v}_{k0}$:

$$\Delta v_k = |\vec{v} - \vec{v}_{k0}| \quad (16)$$

The statistical sampling, which may or may not belong to the $\vec{v}$, with an estimate of the proximity to it (16) is the set of scalar distances estimates of cluster elements (15) from the $\vec{v}_{k0}$

$$\Delta v_k = \{\Delta v_{ki} = |\vec{v}_{ki} - \vec{v}_{k0}| \, i \in [1, n_k]\}. \quad (17)$$

The decision of the vector $\vec{v}$ belonging to the set is accepted for the selected confidence level P_k. To determine the supplies of vector $\vec{v}$ to the sample (17), we need to calculate the average based on a sample, as follows:

$$\Delta\vec{v}_k = \frac{1}{n_k} \times \sum_{i=1}^{n_k} \Delta v_{ki}, \quad (18)$$

The following step requires to compute the standard deviation of the vectors that have already been identified in the cluster:

$$S_{\Delta v_k} = \sqrt{\sum_{i=1}^{n_k} (\Delta v_{ki} - \Delta\vec{v}_k)^2}, \quad (19)$$

and compute the standard average with the sample within the deviation cluster

$$S_{\overline{\Delta v_k}} = \frac{S_{\Delta v_k}}{\sqrt{n_k}}. \quad (20)$$

Further on, and according to the calculated values, it is necessary to calculate the experimental value of one-sample Student t-test criteria:

$$t_{ki} = \frac{|\Delta v_{ki} - \Delta\vec{v}_k|}{S_{\overline{\Delta v_k}}} \quad (21)$$

If the obtained empirical value t_i does not exceed the table value t_p [33] with n degrees of freedom, and can be selected the confidence level P_k in the table, we can assume that $\vec{v}$ belongs to this cluster.

The described algorithm is one of the high quality instruments to study the ME dependencies [33][34]. On its basis, the software tool "EGSO_MET" was developed. The software structure includes 8 separate classes that inherit the standard class Object:

1. Individual class - is used to describe objects such as individual EGA;
2. Cluster class - is used to describe objects such as a cluster;
3. CreatePopulation class - includes methods for creating an initial population of EGA, which consists of a special type of objects;
4. FormPopulation class - contains the methods for the selection and formation of the initial population in EGA based on the user-set parameters;
5. FunctionDeal class - includes methods for calculating the objective function value of the object;
6. EvolutionaryGeneticAlg class - includes methods for simulating crossover and mutation operators in EGA;
7. Student_tTest class - includes methods of clustering obtained with EGA results;
8. MainViewModel class - contains the methods of interaction with the «EGSO_MET» software interface.

A more detailed description of the EGSO_MET interface and structure for each class can be found in [30][31]. This software has an intuitive graphical user interface, which includes a user input settings module, a graphical display of the object in 3-dimensional space and in addition has the statistics gathering module. The instruments used for its reconstruction include:

- Windows Presentation Foundation (WPF) - a system for building the Windows client application with visually appealing possibilities of interaction with the user. The graphics (presentation) subsystem is a part of .NET Framework (since version 3.0), supported with the XAML language.
- Helix toolkit 3D is the graphics framework, based on DirectX engine. It allows you to re-create the elementary 3-dimensional animation.

The input parameters, in addition to the standard input (population size, number of generations, probability of crossover, probability of mutation, the search area, the accuracy, the object under study) have been extended with:

- extremes parameter (minimum / maximum);
- special selection parameter (roulette / casual / tournament);
- parameter of crossover type (single point / two-point);
- parameter of mutation type (one-point / multipoint);
- parameter of breakpoints type.

Intuitive «EGSO_MET» software interface is shown in Fig. 6. On Fig. 6(a) are shown the settings of EGA border parameters, accuracy and extremes type.

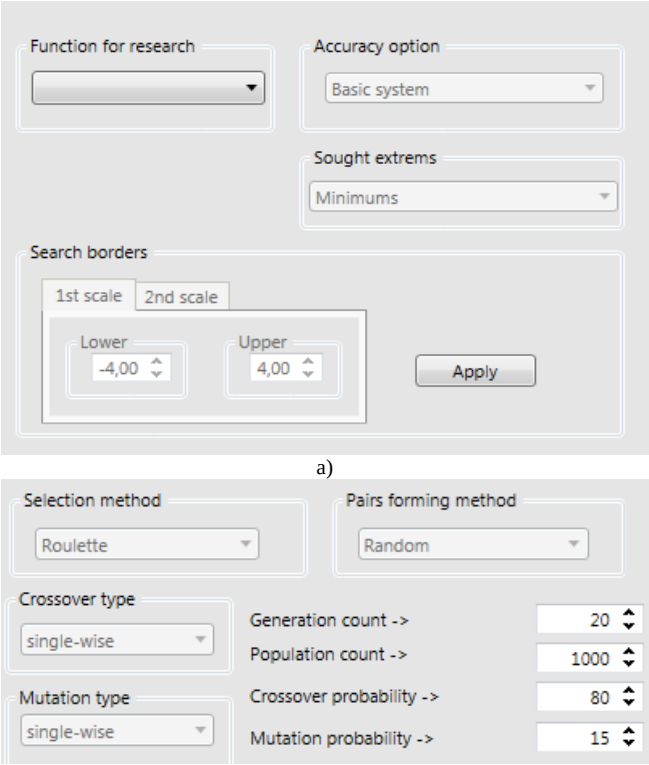


Figure 6. «EGSO_MET» main window settings interface: *a)border settings; b) EGA settings.*

On Fig. 6(a) are shown the settings of EGA border parameters, accuracy and extremes type. On Fig. 6(b) are shown the settings of EGA parameters.

In addition to the tabs of the main window, EGSO_MET software has tabbed settings for the local search options and the tabs for the results to set the global and local search. An example of the displayed results of the global and local searches is shown in Fig. 7.

clusternumber	firstparam	secondparam	functionvalue
1	3.58098	-1.85298	0.00108
2	-3.80655	-3.31459	0.06309
3	2.9888	1.93927	0.07905
4	-2.79521	3.17454	0.07995

Figure 7. «EGSO_MET» obtained results interface.

ME functions research using EGSO_MET software. As example, the modified EGA results of the Himmelblau function research (search minima) and the Shekel function research (maxima search) are presented with the EGSO_MET software described above. It is worth noting that the Himmelblau function study was carried out in the range of [-4, 4], and Shekel function research was carried out in the range [0, 20]. The EGA input parameters are:

- Number of generations = 20;
- Individuals in each generation = 1000;
- Crossover probability = 95%;
- The probability of mutation = 30%;
- The accuracy of the study = 7 digits after the decimal.

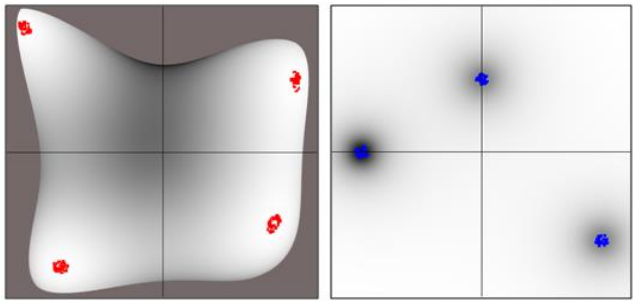
In the study of Himmelblau function, 4 clusters are allocated, and the minimums of each cluster can be correlated with Himmelblau function minimums situated in the study area. In the study of Shekel function, 3 clusters are allocated, and peaks of each cluster can be correlated with the Shekel function peaks situated in the study area.

Figure 3 shows the graphs finding sequentially the values of the Himmelblau function, their various coordinates (X and Y) and the corresponding values of the objective function ($F(X, Y) \approx 0$), which are sorted in the descending order. In Himmelblau function clearly shows that value of the objective function which are close to each other (or in some cases equal to) have significant differences in the coordinate parameters (i.e., parameters of the objective function, providing close to the minimum values are different). This fact confirms that the object has multi-extremes. It should be noted that this property is also inherent to the Shekel function.

In the study of Himmelblau function cluster, the 4 obtained minimum values can be considered for a global extreme, which characterized the local minima of the function, see Fig. 8 (a).

In the study of a Shekel function cluster, see Fig. 8 (b), the 2 obtained extremes are peripheral and their maximum values characterize the local maxima of the Shekel function, and the maximum value of one of the three found with the help of research clusters, characterizes the global Shekel function maximum.

The study of Himmelblau and Shekel functions in finding the global and the local extremes are presented in TABLE II and TABLE III, respectively, with their actual values and their corresponding coordinates.



a) b)

Figure 8. Extremes localization areas: *a) selected clusters of Himmelblau function b) selected clusters of Shekel function.*

As seen from TABLE II and TABLE III, the extremes evaluation values and their coordinates are not very accurate. If the obtained values do not satisfy the required

accuracy, it has to be followed with a second study of the cluster function. As example, the second research in each cluster area of the Himmelblau function is shown in TABLE IV. The authors have developed the approach for the local search in the extreme areas [31], based on the EGA [29][30], where similar extremes values of the 2nd cluster (Himmelblau function) can be seen in Fig. 9 (a); the best extreme evaluation is highlighted with a red circle. The search was carried out in the area around the highlighted extremes, see Fig. 9 (b).

TABLE II. OBTAINED ON THE FIRST ITERATION RESULTS (CLUSTERS OF HIMMELBLAU FUNCTION)

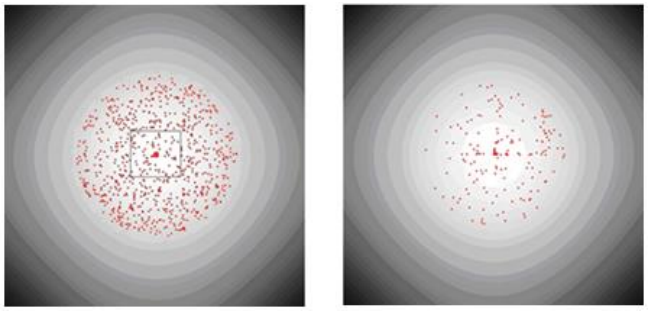
Himmelblau function					
Standard			*Extremal evaluation item*		
X	*Y*	*f(x,y)*	*Coordinates*		*Value*
			X	*Y*	*f(x,y)*
3.58442	-1.84812	0	3.58931	-1.86579	0.00527
3	2	0	2.98944	2.03233	0.01531
-3.77931	-3.28318	0	-3.76109	-3.25452	0.04045
-2.80511	3.13131	0	-2.79797	3.09164	0.06383

TABLE III. OBTAINED ON THE FIRST ITERATION RESULTS (CLUSTERS OF SHAKEL FUNCTION)

Shekel function					
Standard			*Extremal evaluation item*		
X	*Y*	*f(x,y)*	*Coordinates*		*Value*
			X	*Y*	*f(x,y)*
2	10	1.01439	1.94624	9.87414	0.99575
10	15	0.51646	9.95978	14.99479	0.51612
18	4	0.51646	18.08359	4.03641	0.50665

TABLE IV. EXTREMES LOCALIZATION (SECOND RESEARCH) OF HIMMELBLAU FUNCTION

Values	**1st cluster**	**2nd cluster**	**3rd cluster**	**4th cluster**
X	3.58518	2.99946	-3.7783	-2.8023
Y	-1.85084	2.00173	-3.28153	3.1294
f(x,y)	0.00012	0.00004	0.00013	0.0004



a) b)

Figure 9. Form of clusters in localized area: *a) 100th; b) 110th generation.*

C. Solving ME problems by ant colony method.

The "Ant colony method" (ACM) is studied in this section, as the third group of methods that are widely used in solving various optimization problems. A distinctive feature of ACM is that the fundamental behaviour of the real ants is modelled [34][35]. Such behaviour allows the colony to achieve effective results in life, which are often close to optimal solution. As a rule, the ACM is mainly used for the route minimization problems in graph [35][36][37], but, according to studies and to a number of scholars and authors [38][39], these algorithms also show good results in other areas. In this paper, the modification of the classical ACM is used to optimize the reference test ME functions Ursem01 [40] and Styblinski-Tang [41].

In the following section, a method based on the classical implementation of ACM is described. It is used to solve problems on graphs [35], however, for solving the ME problems, some modifications are required.

By analogy with the classical ACM, this modification called MACM, inherited the well-known steps as "placement and initialization", "ants moving" and "breakpoint checking conditions."

Algorithmic and mathematical model of MACM. In general, mathematical and algorithmic model studies the multi-extremal problems which are depending on $\Phi(x)$. It has the arbitrary order n (i.e., x - n - vector), and represented as follows. The field of study $\Phi(x)$ using the MACM in factor space is divided into

$$M = \prod_{i=1}^{n} m_i \qquad (22)$$

fragments – hyper-parallelepipeds, each of which is associated with the value of the function at the centre. Furthermore, each fragment is originally assigned to some small positive pheromone level and a certain number of ants are placed inside the fragment.

Thus, many fragments can be described as a multi-dimensional cellular matrix of the form

$$A = \left(\left(\ldots \left(\left(a_{i_1,i_2} \right)_{i_3,i_4} \right) \ldots \right)_{i_{n-1},i_n} \right), \qquad (23)$$

fulfilling the necessary degree of nesting. The dimension n can be odd. Then the external matrix is column matrix.

The search algorithm implemented in MACO due to [34][35][36][37][40] that every ant in hyper-parallelepiped $a_{i_1,i_2 \ldots i_n}$ evaluates all adjacent hyperparallelepipeds and calculates the probability $P^{i_j \pm 1}_{i_1,i_2,\ldots i_n}$ of transfer expediency according to the (24), where i_i, m_i – the serial number on the fragment location on x_i axis of factor space; Q – the optimization criterion; f – the number of pheromones in a fragment of a particular index; α – the variable pheromone exposure factor on the transition probability of an ant; β - the variable ratio of the intensity variation of the function when passing over the edge.

$$\forall i_1 = \overline{1, m_1}; i_2 = \overline{1, m_2}; \ldots i_n = \overline{1, m_n}; j = \overline{1, n} \rightarrow$$
$$\rightarrow P_{i_1, i_2, \ldots i_n}^{i_j \pm 1} \begin{cases} Q(x_{i_1}, \ldots x_{i_j}, \ldots x_{i_n}) > Q(x_{i_1}, \ldots x_{i_j \pm 1}, \ldots x_{i_n}) \rightarrow \dfrac{\left[f_{i_1, \ldots i_{j \pm 1}, \ldots i_n} \right]^{\alpha} * \left[\Delta Q_{i_1, \ldots i_{j \pm 1}, \ldots i_n} \right]^{-\beta}}{\sum\limits_{j=\overline{1,n}} \sum\limits_{i_k = i_j - 1, i_j + 1} \left\{ \left[f_{i_1, \ldots i_k, \ldots i_n} \right]^{\alpha} * \left[\Delta Q_{i_1, \ldots i_k, \ldots i_n} \right]^{-\beta} \right\}} \\ Q(x_{i_1}, \ldots x_{i_j}, \ldots x_{i_n}) < Q(x_{i_1}, \ldots x_{i_j \pm 1}, \ldots x_{i_n}) \rightarrow 0, \end{cases} \quad (24)$$

Model (24) is supplemented by model updates pheromone - the main tool of giving an effective search, inherent only to ACM. Its essence, at each iteration, occurs both the growth and the evaporation of pheromone. Therefore, changing the pheromone stock in each fragment $a_{i_1, i_2 \ldots i_n}$ in one simulation step h is calculated by the following equation of state in discrete form:

$$f_{a_{i_1, i_2 \ldots i_n}}(h+1) = (1-\rho) * f_{a_{i_1, i_2 \ldots i_n}}(h) + \Delta f_{a_{i_1, i_2 \ldots i_n}}(h) \quad (25)$$

where: $\rho \in (0; 9)$ – the variable evaporation coefficient; $f_{a_{i_1, i_2 \ldots i_n}}(h)$ - pheromone content in $a_{i_1, i_2 \ldots i_n}$ hyper-parallelepiped; $\Delta f_{a_{i_1, i_2 \ldots i_n}}(h)$ – the increment on each iteration, calculated according to the formula:

$$\Delta f_{a_{i_1, i_2 \ldots i_n}}(h) = K * \left(Q(x_{i_1}, \ldots x_{i_j \pm 1}, \ldots x_{i_n}) - Q(x_{i_1}, \ldots x_{i_j}, \ldots x_{i_n}) \right) \quad (26)$$

where K – the pheromone growth coefficient.

The phenomenon of pheromone evaporation is taken as real property information exchange and causes the ant to confirm or update its results within the search model, thus providing a review of the whole space of possible solutions.

When looking for the minima where $Q\left(x_{i_1}, \ldots x_{i_j}, \ldots x_{i_n}\right) < Q\left(x_{i_1}, \ldots x_{i_j \pm 1}, \ldots x_{i_n}\right)$ is satisfied, transition between fragments is banned. Thus, the breakpoint condition is fulfilled if all the ants are unable to move. As a result, after N iterations ants get fragment with the lowest functions value and localize the minimums.

The software tool Multi-Extreme Optimization of Function by MACM description. On the basis of the described algorithm and model (22)-(26), the software tool (ST) was developed that implements the search of local and global extremes. The ST structure includes 6 independent classes that inherit from the standard class Object:

1. class *Ant* – class that is used to describe objects such as ant;
2. class *Algorithm* – class that is used to describe objects of MACM algorithm;
3. class *Drawing* – class that contains methods for GUI;
4. class *Parameters* – class that contains global parameters;
5. class *Results* – class that is used to generate and output the resultant information;
6. default classes *Form* and *Program* – standard classes that are created by default in the development environment.

ST has an intuitive graphical user interface, which includes a user input settings module, a graphical display of the object, as well as statistics collection and displaying the results modules. To create a modelling module were involved:

- Windows Forms — Application Programming Interface (API), is responsible for the graphical user interface and is part of Microsoft.NET Framework;
- Tao Framework — a library that provides developers with .NET and Mono access to features of popular libraries like OpenGL and SDL.

Fig. 10 shows the software tool graphical user interface (GUI).

The user settings are located on the right side of the the main GUI window, and in the left side of the visual represenattion of the object is dispalyed. The graphical display is based on the Tao framework library using OpenGL. The settings window allows to change all the parameters of the algorithm in an easy way. In the process of implementation, all the information is gathered and displayed on the screen for the visual assessment of the computed optimization results.

To display all localized extremums and their status, a tab with the results is shown in Fig. 11.

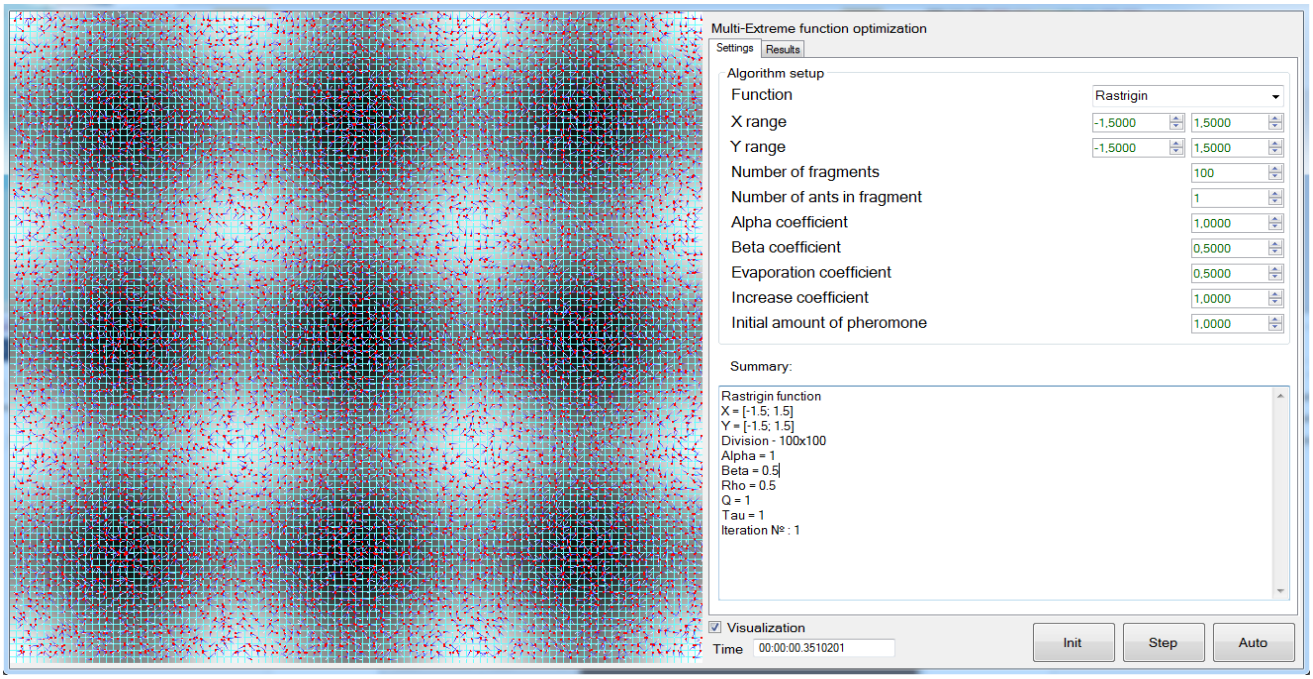
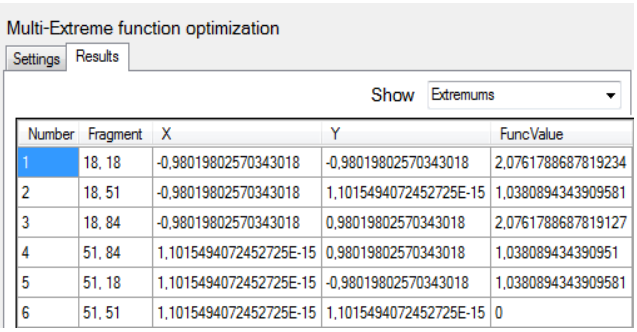


Figure 10. «MEOF_MACM» main GUI

ME functions research using MEOF_MACM. As example, we considered and optimized Ursem01 function and Styblinski-Tang, which plots are presented in Fig. 12.

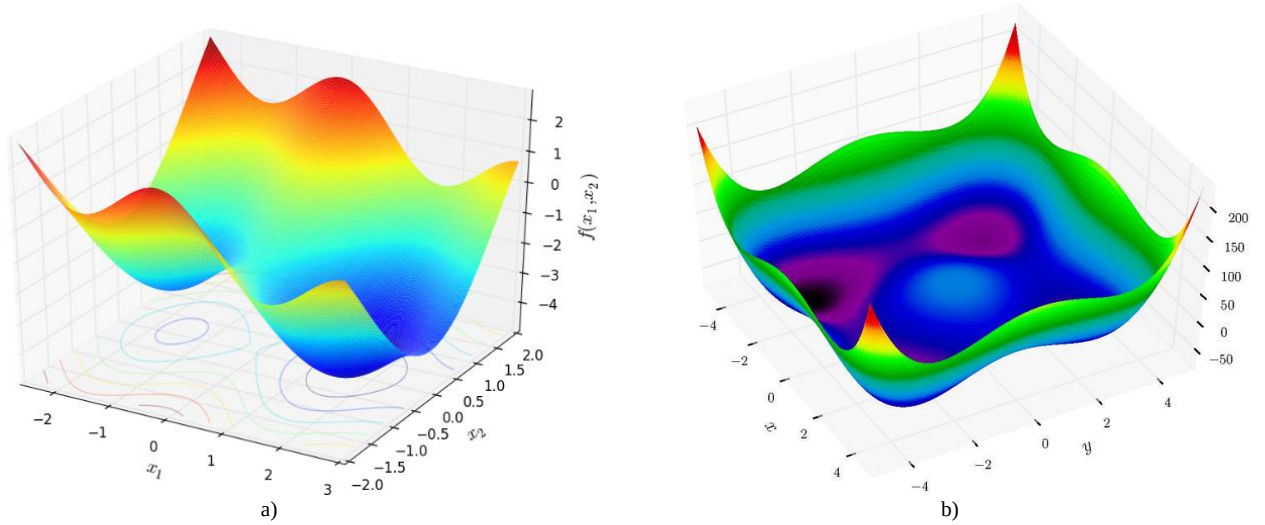
Research Ursem01 function is performed in the range of x and y [-2;2] coordinates. The selected area is initially divided into fragments with 0.0133 step, and one ant was placed on each fragment. Coefficients $\alpha = 1$, $\beta = 0.7$, $\rho = 0.5$, $K = 1$ and $\tau = 1$. Fig. 14 shows the individual stages of ST.

The localization results of each extrema are presented in TABLE V.



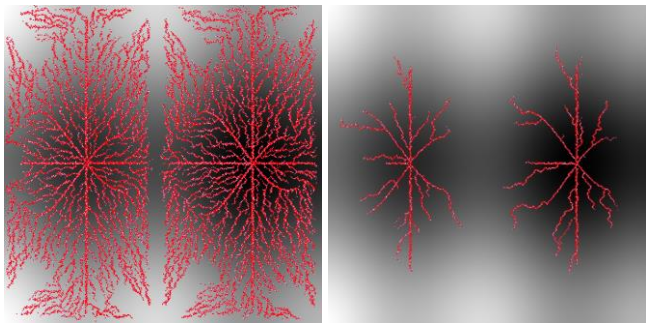
Number	Fragment	X	Y	FuncValue
1	18, 18	-0,98019802570343018	-0,98019802570343018	2,0761788687819234
2	18, 51	-0,98019802570343018	1,1015494072452725E-15	1,0380894343909581
3	18, 84	-0,98019802570343018	0,98019802570343018	2,0761788687819127
4	51, 84	1,1015494072452725E-15	0,98019802570343018	1,038089434390951
5	51, 18	1,1015494072452725E-15	-0,98019802570343018	1,0380894343909581
6	51, 51	1,1015494072452725E-15	1,1015494072452725E-15	0

Figure 11. «MEOF_MACM» result interface



a) b)

Figure 12. Plots of additional functions to study (a – Ursem01 function plot, b – Styblinski-Tang function plot)



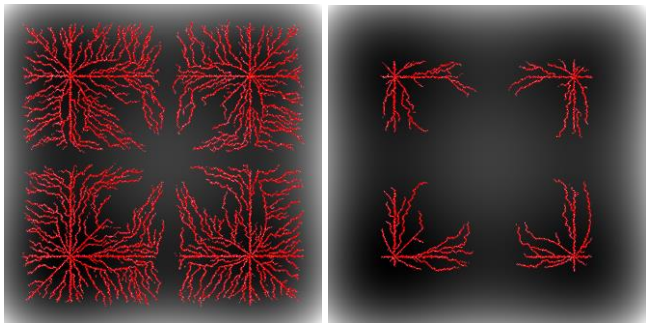
a) b)

Figure 13. The stages of the localization 300x300 division area (a, b – intermediate results)

TABLE V. LOCALIZED RESULTS (URSEM01 FUNCTION)

Ursem01 function					
Standard			**Extremal evaluation item**		
X	*Y*	*f(x,y)*	*Coordinates*		*Value*
			X	*Y*	*f(x,y)*
1.69714	0	-4.8168	1.69500	-0.00499	-4.81676
-	-	-	-1.44666	-0.00666	-3.24594

The research on the Styblinski-Tang function is performed in the range [-5; 5] with the partition of the test area by 400x400. Fig. 14 shows the individual stages of ST.



a) b)

Figure 14. The stages of the localization 400x400 division area (a, b – intermediate results)

The localization results of each extremum are presented in TABLE VI. If the results are not accurate enough, the algorithm can be repeated with the resulting fragments, and in such way the extreme will be found.

TABLE VI. LOCALIZED RESULTS (STYBLINSKI-TANG FUNCTION)

Styblinski-Tang function					
Standard			**Extremal evaluation item**		
X	*Y*	*f(x,y)*	*Coordinates*		*Value*
			X	*Y*	*f(x,y)*
-2.90353	-2.90353	-39.1659	-2.91249	-2.91249	-39.16477
-	-	-	-2.91249	2.73749	-32.09647
-	-	-	2.73749	-2.91249	-32.09647
-	-	-	2.73749	2.73749	-25.02818

It is worth noting that this algorithm and its software implementation have no special mechanism for clustering. This is due to the fact that the clustering mechanism is incorporated in the mathematical models (22). This approach is characterized by some discreetness. The test area is divided into fragments, thus the resulting agents somehow are combined into groups, which are further referring to a specific function value within the fragment.

D. Computational resources and performance

The search of the extremes by the swarming particles, evolutionary-genetic and ant colony algorithms on 2-dimensional Rastrigin function is carried out on a PC with processor AMD Phenom II P960 with 6 GB of RAM.

To achieve the accuracy 10^{-3}, the time was up to 40 sec. For the additional search within each area, the required computational time was up to 20-50 sec.

III. RELATED WORK

In the design optimization process, we are often confronted with problems facing the ME conditions. Such situation requires several decisions to be taken, which take into consideration several identical or close extremes, and the best choice in-between them has to be used. The classical theory of scheduling gives examples, where several identical optimums and identical sub-optimums, close to them exist. The majority of discrete, integer and combinatory programming problems differs in such property, in particular, when finding solution for graphs. The finite number (though to be very big) of admissible decisions requires considering the ME solutions for the discrete environment optimization. It is important to have a complete solution of the ME task, because the criterion is usually a numerical expression related to the optimized object. However, there are many additional conditions, which can help to choose the extreme, equivalent or close in size, and satisfy both, the numerical criteria estimates and the heuristic ideas. Therefore, the choice of the most effective methods and algorithms, is an extremely important step to find such solution of the ME task.

However, not all of the search methods provide the successful solution for the ME task. It is well known that the determinate methods are sensitive to the sign-variable, so-called "gullied" surfaces, which define the real variables in the factor space. The solution of discrete tasks by such methods leads to the nondeterministic polynomial, in order to be defined for the complete problem in time. The methods of the accidental search are poorly predictable, since it is impossible to control the time expenditure, and even the basic decision, on which heuristic method to apply, when having a real search optimization problem. In particular, in Russia, in the last years, the quite intensive research is conducted to find appropriate solutions for the many optimization problems. Among these methods, it is important to mention the swarming particles algorithm [14][15][16][17][18][19][20][21], the evolutionarily genetic algorithm [25][26][27][28][29][30][31][32][33], and the ant colony algorithm [34][35][36][37][38][39][40][41]. These algorithms were investigated, as the traditional optimization tasks, and in relation to find the solution of the ME tasks. For the last case, they have been significantly modified, by

experimenting with different heuristic methods, which research was already conducted by the authors. Therefore, the presented work brings forward a peculiar theoretical result, and trace the roadmap for the future research in this direction.

IV. CONCLUSION

The application analysis of the 3 heuristic algorithms for solving the ME tasks showed that these methods are efficient, effective, and bring some essential features to the presented solutions.

The specific approaches to solve the task for each of these particular cases is determined through the analysis of the algorithms features; the detection and identification of local extremes, clustering methods and subsequent operations resulting from such analysis. However, in all these cases, the modifications of algorithms is connected with the data clustering necessity, which was found to be essential. In addition, all the methods showed reasonable performance.

To conclude, all the 3 studied methods are considered to be relevant and promising for the future applications. The specific choice of the algorithm tool for solving ME tasks depends on the experience and personal researcher preferences, as well as on the special features of the domain specific research area.

In this paper, the task of finding the set of extremes for 2-dimensional Rastrigin test function was examined. In future research, it is advisable to study the problem of higher dimension (3 or more) in order to assess the impact of algorithms' parameters affecting the time and search accuracy, and to enable algorithms modifications for the mathematical models of any-scale problem dimension.

REFERENCES

[1] R. Neydorf, I. Chernogorov, V. Polyakh, O. Yarakhmedov, J. Goncharova, and D. Vucinic "Study of Search Optimization Opportunities of Heuristic Algorithms for Solving Multi-Extremal Problems," Proceedings of The X International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP), 2016, pp. 44-51.

[2] S. Boettcher and A. G. Percus, "Extremal optimization: methods derived from co-evolution," Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO), 1999, pp. 825-832.

[3] C. A. Floudas and P. M. Pardalos, "Encyclopedia of optimization, 2nd Edition," Springer, New York: Springer Scince+Business Media, LCC, 2009.

[4] K. B. Jones, "Search engine optimization, 2nd edition," Indianapolis: Wiley Publishing, 2010.

[5] R. Shreves, "Drupal search engine optimization," Birmingham: Packt Publishing LTD, 2012.

[6] I. M. Vinogradov, "Mathematical encyclopedia," Soviet Encyclopedia, vol.4, 1977-1985, pp. 135-140.

[7] R. G. Strongin, "Algorithms for multi-extremal mathematical programming problems," 1992, pp. 357-378.

[8] R. A. Neydorf, A. V. Filippov, and Z. H. Yagubov, "Commute algorithm of biextreme solutions of the homogeneous distribution problem," Herald of DSTU, №5(56), vol.11, 2011, pp. 655-666.

[9] R. A. Neydorf and A. A. Zhikulin, "Research of properties solutions of multi distribution problems," System analysis, management and information processing: Proceedings of the 2nd International Scientific Seminar, Rostov-on-Don: IC DSTU, 2011, pp. 377-380.

[10] R. A. Neydorf and A. A. Dereviankina, "The methodology of solving problems of the modified method of multi swarming particles," Innovation, ecology and resource-saving technologies at the enterprises of mechanical engineering, aviation, transport and agriculture, Proceedings of the IX International Scientific and Technical Conference, Rostov-on-Don: IC DSTU, 2010, pp. 328-330.

[11] R. A. Neydorf and A. A. Dereviankina, "Decision of multi tasks by dividing swarms," Herald of DSTU, №4(47), vol.10, 2010, pp. 492-499.

[12] R. A. Neydorf and A. A. Dereviankina, "The solution of problems of recognition by swarming particle swarm division," News of SFU. Technical science. Special Issue (Intellectual CAD), Taganrog: Publisher TTI SFU, №7(108), 2010, pp. 21-28.

[13] L. A. Rastrigin, "Systems of extremal control," Nauka, Moscow (in Russian), 1974.

[14] R. C. Eberhart and J. Kennedy, "New optimizer, using particle swarm theory," Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, 1995, pp. 39-43.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of IEEE International Conference on Neural Networks IV, 1995, pp. 1942-1948.

[16] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, New Jersey, 1998, pp. 69-73.

[17] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multi-dimensional complex space," IEEE Transactions on Evolutionary Computation, 2002, pp. 58-73.

[18] Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," Evolutionary Computation, IEEE Transactions on 8(3), 2004, pp. 204-210.

[19] R. A. Neydorf and I. V. Chernogorov, "Parametric configuration of the algorithm of searching optimization by swarming particles using experimental planning," International Institute of Science "Educatio", №2(9), vol.4, 2015, pp. 44-49.

[20] R. A. Neydorf and I. V. Chernogorov, "Increased functionality of the method of swarming particles by kinematic and dynamic modification of the algorithm of its realization," LTD "Aeterna", International Journal "Innovative science", №6, vol. 1, 2015, pp. 24-28.

[21] R. A. Neydorf and I. V. Chernogorov, "A parametric research of the algorithm of swarming particles in the problem of finding the global extremum," Mathematical methods in technique and technologies – (MMTT-28): Proceedings XXVIII International Scientific Conference, YA.6, Saratov: SSTU, 2015, pp. 75-80.

[22] A. A. Barsegjan, M. S. Kuprijanov and V.V. Stepanenko, "Methods and analysis models of OLAP and Data Mining," Saint Petersburg: BKhV-Peterburg, 2004, 336 p.

[23] https://habrahabr.ru/post/124978 (December 15, 2016).

[24] https://www.sfu.ca/~ssurjano/schwef.html (December 15, 2016).

[25] A. Fraser, "Computer models in genetics," New York: McGraw-Hill, 1970.

[26] D. Goldberg, "Genetic algorithms in search, optimization and machine learning," Addison Wesley, 1989.

[27] H. Mühlenbein, D. Schomisch, and J. Born, "The parallel genetic algorithm as function optimizer," Parallel Computing, vol. 17, 1991, pp. 619-632.

[28] N. A. Barricelli, "Esempi numerici di processi di evoluzione," Methodos, 1954, pp. 45–68.

[29] S. Boettcher, "Extremal optimization - heuristics via co-evolutionary avalanches," Computing in Science & Engineering 2, 2000, pp. 75–82.

[30] S. Boettcher, "Extremal optimization of graph partitioning at the percolation threshold," 1999, pp. 5201–5211.

[31] R. A. Neydorf and V. V. Polyakh, "Method of multisearch using evolutionary genetic algorithm and sample t-test," LTD "Aeterna", International Journal "Innovative science", №3, vol.1, 2015, pp. 135-140.

[32] R. A. Neydorf and V. V. Polyakh, "Study of multi dependencies using an evolutionary genetic method and one sample Student's t-test," Mathematical methods in technique and technologies – (MMTT-28): Proceedings XXVIII International Scientific Conference, YA.6, Saratov: SSTU, 2015, pp. 83-87.

[33] R. A. Neydorf and V. V. Polyakh, "Localization search scopes evolutionary genetic algorithm for solving problems of multi nature," Science.Technology.Production, №6, vol.2, 2015, pp. 18-22.

[34] M. Lovric, "International encyclopedia of statistical science," Springer-Verlag Berlin Heidelberg, 2011.

[35] Bert Holldobler. "The Superorganism: the beauty, legance and strangeness of insect societies"./ Bert Holldobler, Edward O. Wilson. // 2008: W.W. Norton and Company, New York, 576 pp., 110 color and 100 black-and-white illustrations. ISBN-10: 0393067041, ISBN-13: 978-0393067040, Price: Euro 41.99.

[36] R. A. Neydorf and O. T. Yarakhmedov, "Development, optimization and analysis of parameters of classic ant colony algorithm in solving travelling salesman problem on graph," Science. Technologies. Production, №3, vol.2, 2015, pp. 18-22.

[37] A. Kazharov and V. Kureichik, "Ant colony optimization algorithms for solving transportation problems," Journal of Computer and Systems Sciences International, №1, vol.49, 2010, pp. 30–43.

[38] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, №1, vol.1, 1997, pp. 53-66.

[39] X. Liu and H. Fu, "An effective clustering algorithm with ant colony," Journal of Computers, №4, vol.5, 2010, pp. 598-605.

[40] M. D. Toksari, "Ant colony optimization for finding the global minimum," Applied Mathematics and Computation 176, 2006, pp. 308–316.

[41] Jani Rönkkönen, "Continuous Multimodal Global Optimization with Differential Evolution-Based Methods," Ph.D. Thesis, Lappeenranta University of Technology, Lappeenranta, Finland, 2009, [Accessed May. 7, 2015]. [Online].Available:https://www.doria.fi/bitstream/handle/10024/50498/isbn%209789522148520.pdf

[42] Global Optimization Test Functions Index. Retrieved June 2013, from http://infinity77.net/global_optimization/test_functions.html#test-functions-index.

[43] Michael L. Pinedo "Scheduling Theory, Algorithms, and Systems Fourth Edition," ISBN 978-1-4614-1986-0 e-ISBN 978-1-4614-2361-4 DOI 10.1007/978-1-4614-2361-4 Springer New York Dordrecht Heidelberg London Mathematics Subject Classification (2010): Library of Congress Control Number: 68Mxx, 68M20, 90Bxx, 90B35.

[44] https://www.encyclopediaofmath.org/index.php/Discrete_programming (September 29, 2016).

[45] Donald E. Knuth – "The Art of Computer Programming. " vol. 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions (vi+240pp, ISBN 0-321-53496-4).

[46] http://www.cs.utsa.edu/~wagner/knuth/ (September 29, 2016).

[47] GRhttp://diestel-graph-theory.com/index.html (September 29, 2016).

[48] Keijo Ruohonen. "Graph theory," (Translation by Janne Tamminen, Kung-Chung Lee and Robert Piché), 2013.

[49] Christopher Griffin, "Graph Theory: Penn State Math 485 Lecture Notes Version" 1.4.2.1 2011-2012.

[50] Paul Van Dooren "Graph Theory and Applications," Université catholique de Louvain Louvain-la-Neuve, Belgium Dublin, August 2009.