Tourist Itineraries Plan Design Based On the Behavior of Bee Colonies

Dmytro Uhryn[1][0000-0003-4858-4511], Oleh Naum[2][0000-0001-8700-6998], Nataliya Antonyuk[3][0000-0002-6297-0737], Ivan Dyyak[4][0000-0001-5841-2604], Liliya Chyrun[5][0000-0003-4040-7588], Andriy Demchuk[6][0000-0001-6942-9436], Victoria Vysotska[7][0000-0001-6417-3689], Zoriana Rybchak[8][0000-0002-5986-4618], Taras Batiuk[9][0000-0001-5797-594X]

[1]Chernivtsi Philosophical and Legal Lyceum, Chernivtsi, Ukraine
[2]Drohobych Ivan Franko State Pedagogical University, Drohobych, Ukraine
[3-4]Ivan Franko National University of Lviv, Lviv, Ukraine
[3]University of Opole, Opole, Poland
[5-9]Lviv Polytechnic National University, Lviv, Ukraine

ugrund38@gmail.com[1], oleh.naum@gmail.com[2], nantonyk@yahoo.com[3], ivan.dyyak@lnu.edu.ua[4], lchirun21@gmail.com[5], Andrii.B.Demchuk@lpnu.ua[6], Victoria.A.Vysotska@lpnu.ua[7], zozylka3@gmail.com[8]

Abstract. The article deals with the modified paradigm of the bee colony for tourist routes by solving combinatorial problems on graphs: selection in the graph of an independent subset of vertices, finding the maximum pairing in the graph, coloring the graph, highlighting the click in the graph. Based on the analysis of the behavioral model of bee colony self-organization, methods and mechanisms of formation of corresponding representations of solutions of the considered combinatorial problems on graphs are developed. Methods of search space forming are considered. The position in the search space is presented as an ordered list. The key operation of the bee algorithm is the study of perspective positions and their surroundings in the search space. The paper proposes a method of decision edges forming with an adjustable degree of similarity and closeness between them. Three approaches are proposed to determine the number of foraging agents that are sent to the vicinity of each base position.

Keywords: Bee Colony, Tourist Routes, Optimization, Bee Algorithm, Self-Organization.

1 Introduction

In the tourism sector, an interest of every tourist in visiting all the tourist units is to solve a number of simultaneous tourist requests. The most relevant requirements which are to be met by the terms of a tourist travel include: a calendar planning of tourist trips, travel routes which can cover maximum tourist sites with the minimum distances between them; providing the necessary set of services depending on the type of tourism

activity. The search of optimal routes with minimum distances and coverage of the maximum number of tourist offers is a real problem of the entire tourism industry. One of the most pressing and at the same time the most difficult issues of final measurement of continuous optimization in practical terms is the problem of global optimization of conventional tourist routes. Solutions to problems of this sector can divided into two areas of methods:

- methods for reducing the issues of global conditional optimization to the issues of global unconditional optimization using penalty or barrier functions;
- methods which are specifically designed to solve the issue of global conditional optimization.

The method of bee colonies behavior, considered in this article, refers to methods of the first group. This method is designed to solve the problems of global unconditional optimization.

Methods to resolve the problem of unconditional global optimization are divided into deterministic, stochastic and heuristic [1].

Heuristics methods are relatively new and are rapidly developing. Evolutionary and behavioral (imitation) methods should noted among these methods.

Behavioral methods of solving the problem of global unconditional optimization are based on modeling the collective behavior of self-organized living or inanimate systems. The interacting elements of these systems are usually called agents. Key ideas of behavioral methods include decentralization, agent interaction, and simplicity of agent behavior. In other words, such methods have a bionic nature for living systems, which means they are based on modeling the behavior of insects, birds, animals, etc., whose behavior is collective in nature, due to which the so-called collective intelligence is achieved.

The main feature of multi-agent methods of collective intelligence is their bionic nature - the colonies analysis methods for solving optimization problems, including the evolutionary optimization method (including genetic algorithms), methods of ant and bee colonies. It should noted that these methods model the behavior of different groups of social animals, insects and other creatures. This allows these groups to solve various difficult practical problems in nature, indicating the effectiveness of their behavior and, consequently, the effectiveness of these methods.

2 Objects and Methods

When implementing these methods there was used an agent-oriented programming paradigm, which is based on the modeling of collective intelligence and includes: the method of Ant Colony Optimization (ACO), a method of Bee Colony optimization (BCO), Particle Swarm Optimization (PSO) and other methods. These methods are already effectively used to solve different problems: ACO is used to solve the traveling salesman problem, issues of scheduling, feature selection, clustering, etc.; BCO - to solve the problem of calendar planning, solving the problem of the salesman, solving the transport problem and others [2].

3 Problem Statement

To solve the problem of finding optimal routes, compiling calendar schedules of tourist transport with the provision of maximum service on the example of collective intelligence, it is necessary to determine the functions performed by social insects in the process of solving various problems. The bee colony method is a heuristic iterative method of random search [5]. It used to solve various optimization problems that relate to both discrete and continuous optimization, which will improve the quality of tourism plan development.

4 Analysis and Discussion

To describe the behavior of bees in nature, three basic concepts are used: the source of nectar (flower), busy worker bees (foragers), free worker bees (scouts). The source of nectar is characterized by its usefulness, which is determined by such factors as distance from the hive, the concentration of nectar, the convenience of its production.

Busy worker bees are bees that are "associated" with one of the sources of nectar, i.e. extract nectar from it. Foragers have the following information about "their" source of nectar: the direction from the hive to the source and the usefulness of the source.

Free worker bees are scout bees that search for nectar sources for use, as well as observer bees that are currently doing some work in the hive.

Dancing is a message about the appearance of a source of nectar and pollen, the discovery of water resources or a new place suitable for housing, etc. The scout bee, having found a rich source of nectar, after returning to the hive dances on honeycombs with a full beak of prey. In the dance, the bee also indicates the direction in which to fly to the source, relative to the sun. Even in cloudy weather, bees navigate by the sun.

If a bee decides to leave the hive to get nectar, it flies after one of the scout bees to a place with nectar. Thus, the freebee becomes busy. The mechanisms by which it decides to follow another bee are not well studied, but it is assumed that the recruitment among bees from a mathematical point of view is always a function of the quality of the nectar source. After reaching the place with nectar, busy worker bees extract nectar and return to the hive, leaving the nectar there. After the bee leaves the nectar, it can perform one of the following three actions [5]: to leave the source of nectar and become a free worker bee again; to continue to fly to that source of nectar without recruiting other bees of its hive; to perform a dance and thus recruit other bees. The bee chooses one of the alternatives with some probability [5]. Thus, the functions between busy bees and scout bees are divided into the improved study of found places with nectar and finding new places with nectar, respectively. This division of responsibilities results in the efficient operation of the entire swarm of bees. Thus, the self-organization of a bee swarm is based on the following four main mechanisms:

1. A positive feedback: on the basis of information received from other bees, a bee flies to a source of nectar;

2. A negative feedback: on the basis of information received from other bees, the bee can decide that "its" source of nectar is much worse than other sources and leave this source;
3. A chance event: probabilistic search by scout bees for new sources of nectar;
4. The multiplicity of interactions: information about the source of nectar found by one bee and transmitted to many other bees in the hive.

Based on the proposed approach the bee colony method Bee Colony Optimization for Job-Shop Scheduling Problem (BCO-CHAP) is developed. The scheduling problem can characterized by a set of jobs, each of which consists of one or more operations. Operations are performed on a specific sequence of special machines. The purpose of planning is to schedule a job that minimizes (maximizes) the measure of performance. The scheduling problem refers to NP complex. The measure of performance includes:

- Load of tourist transport (tourist transport utilization rate);
- Route cycle time;
- Performance (cost, throughput);
- A level of stocks and services.

In general, the scheduling problem is represented by a disjunctive graph. The graph consists of nodes that represent operations. There are also two additional nodes that make up resources and costs. A set of oriented arcs is used to describe the benefits of each job. Since the main features of the bee colony method are vulturing dance and foraging process [5], the proposed modification for solving the scheduling problem differs in these stages of the bee colony method in comparison with the previously proposed methods. An analogy of the source of nectar in this modification is a route that can considered as a solution to the scheduling problem.

Upon returning to the hive, the agent performs a swaying dance with probability p. Duration D_i the swaying dance of the *i-th* agent is calculated by the formula: $D_i = d_i \cdot A,$ where A is scaling coefficient, d_i is the relative usefulness of the found source of nectar of the *i-th* agent. The absolute usefulness of the nectar source of the *i-th* agent Pf_i for the scheduling problem is calculated as follows:

$$Pf_i = \frac{1}{C_i},$$

where C_i is a target function for the *i-th* agent path. In this case, it represents the duration of all job operations for the path [5].

Then, having calculated the absolute usefulness of each agent, it is possible to get the average utility of the whole colony Pf_{colony} [5]:

$$Pf_{colony} = \frac{1}{n}\sum_{j=1}^{n} Pf_j,$$

where n is number swaying dances performed at time t.

Thus, we can calculate the relative usefulness d_i for i-th forager [5]:

$$d_i = \frac{Pf_i}{Pf_{colony}}.$$

The probability p_i that the *i-th* agent, after performing the dance, will be followed by other unoccupied foragers, is calculated as follows [3, 4]:

$$p_i = \begin{cases} 0{,}60, if\ Pf_i < 0{,}9 \cdot Pf_{colony}; \\ 0{,}20, if\ 0{,}9P \cdot f_{colony} \le Pf_i < 0{,}95 \cdot Pf_{colony}; \\ 0{,}02, if\ 0{,}95 \cdot Pf_{colony} \le Pf_i < 1{,}15 \cdot Pf_{colony}; \\ 0{,}00, if\ 1{,}15 \le Pf_i. \end{cases}$$

Since in the process of foraging agents form solutions by moving from node to node on a graph describing possible jobs, it is necessary to calculate the probability of adding a given node to the agent path. The probability P_{ij} that the agent chooses the next *j*-th node, while being in the *i*-th node, is calculated as follows:

$$P_{ij} = \frac{\rho_{ij}^{\alpha} \cdot d_{ij}^{-\beta}}{\sum_{j \in J^k} \rho_{ij}^{\alpha} \cdot d_{ij}^{-\beta}},$$

where ρ_{ij} is cost of the arc between *j*-th and *i*-th nodes; d_{ij} is a heuristic distance between *j*-th and *i*-th nodes; $\alpha, \beta \in [0;1]$- factors chosen experimentally; J^k is a set of nodes to which it is possible to move from the *i*-th node. The score ρ_{ij} is determined by a formula [5]:

$$\rho_{ij} = \frac{1 - m\alpha}{k - m},$$

where *k* is the number of nodes to which it is possible to move from the *i*-th node; *m* is the number of path advantages, which may be equal to 1 or 0 [5]. The best way is the one that at any iteration is considered suitable for performing the dance. The number of so-called elite ways is limited. Thus, on the initial iteration all edges have a number *m* = 0, which makes the chances of choosing any edge equal [5].

5 Setting Combinatorial Problems on Graphs

The pairwise combination of a graph $G = (X, U)$ is a subset of such edges $U^* \subset U$, where any two edges $u_k, u_l \in U^*$ do not have common vertices, i.e. are adjacent. A maximum power pair is defined as a pair that includes the maximum number of edges [5, 6], $|U|^* = \max$. Let us build a graph $G_d = (U, V)$ which is dual for graph *G*. Vertices of the graph G_d correspond to the edges of the graph *G*. The pair of vertices (u_i, u_j) in the graph G_d are connected by an edge v_k if and only if in the graph *G* the corresponding pair of edges (u_i, u_j) are adjacent, i.e. incident to one vertex. The set $X_0 \subset X$

of graph vertices $G=(X,U)$ is called internally stable if any two vertices $x_i \in X_0$ and $x_j \in X_0$ are not adjacent. The maximum number of nodes in internally stable set of graph *G* is called the number of internal stability and is designated $\alpha(G)$. Sometimes the number of internal stability is also called the independence graph number *G*.

Thus, the pairwise connection in the graph G corresponds to the intrastable subset of the dual graph G_d. The maximum power pair in the graph G corresponds to the boundary intrastable subset (containing the largest number of vertices) of the dual graph G_d.

Coloring a graph is the assignment of colors to its vertices in such a way that no two adjacent vertices are marked with the same color. [7] The minimum number of colors in which it is possible to color a graph G is called a chromatic number and is denoted $\chi(G)$. If in the graph *G* to select *s* disjoint internally stable subsets of vertices, then the graph can be painted in *s* colors. In other words, the problem of coloring the graph is reduced to the problem of forming in the graph G disjoint intrastable subsets of vertices. The clique of a graph *G* is the maximum set of vertices of the graph X_0, any two of which are adjacent. Let $G_n=(X,U_n)$ be a full graph, built on a set of vertices *X*. Graph $G_k=(X,U_k)$ is the complement graph for $G=(X,U)$ when $U_k=\frac{U_n}{U}$ that is $U_n=U_k \cup U$. It is easy to see that during the transition from a graph *G* its complement G_k each clique in *G* becomes an independent set in G_k. Hence, the task of allocation a clique in the graph *G* is reduced to the task of selection of an independent set of vertices in the graph G_k that is the complement graph for *G* [8, 9].

There is the graph $G=(X,U)$, where *X* is the set of vertices $G=(X,U)$ *U* is set of edges. Let us formulate the task of forming a graph $G=(X,U)$ of intrastable set of vertices $X_1 \subset X$ as a problem of breakdown. It is necessary to break the set *X* into two non-empty disjoint subsets X_1 and X_2 such that any two vertices $x_i \in X_1$ and $x_j \in X_1$ are not contiguous, $X_1 \cup X_2 = X$, $X_1 \cup X_2 = X$. Let $|X_1|=n_1$, $|X_2|=n_2$, $n_1+n_2=n$ [2]. Optimization criterion is the number of vertices $F=n_1$ in a subset X_1. The purpose of the optimization is maximizing of the criterion *F*.

After the formation in the graph $G=(X,U)$ of an intra-stable set of vertices $X_1 \subset X$ to build a pairing or selection in the clique column, the transition from graph *G* to the original graph G_{out} is performed. In this case, when constructing a pairwise connection, the graph *G* is considered as dual to the original graph G_{out} and when selecting a clique, the graph *G* is considered as additional to the original graph G_{out}. When solving the problem of coloring the graph, the subset X_1 is colored in one color and is excluded from *X*. Then the same steps are followed until all the vertices are colored.

6 Presenting Solutions in Algorithms Based on Bee Colony

The first task in developing an algorithm based on the paradigm of bee colonies is to build search space. The position of the search space a_s is represented as an ordered list $E_s = \{e_{si} | i = 1,...,n\}$ of the numbers of the graph vertices G, where n is the number of vertices. List E_s is actually a source solution. The formation of the corresponding list E_s of solutions - intrastable set X_s is carried out step by step by sequentially reviewing the elements of the list E_s, starting with the first. At each step i of the view there is a list of vertices that are already included in the set that is formed $X_S(i)$ where $X_S(1) = \varnothing$. The next vertex is considered e_{si}. If among the vertices of the $X_S(i)$ set there is no vertex adjacent to the vertex e_{si}, then e_{si} is included in $X_S(i)$. As a result of performing this sequential procedure, an intrastable set X_S and a list of other vertices E_{os}, formed by removing the vertices X_S of the set E_s are formed. Let's call the ordered list E_{os} a remainder. Thus, the position of the search space a_s represented as an ordered list E_s corresponds to the intra-stable set X_S and the remainder E_{os}. The estimate of the position a_s is the estimate of the set X_S. We will call the three parameters E_s, X_S, E_{os} the position parameters of a_s.

The key operation of the bee algorithm is the study of promising positions and their neighborhood in the search space. Let us focus on the concept of neighborhood. The importance that is originally embedded in the concept of the neighborhood is that the solutions that lie in the neighborhood of a position, has a high degree of similarity and, as a rule, slightly differs from it. The paper proposes a method of forming neighborhood decisions with an adjustable degree of similarity and closeness between them.

Consider the principles of formation of the position a_z, located in the neighborhood of the base position $a^{\delta}{}_s$. The λ elements are randomly removed from the formed set X_s. The set $X_s(\lambda)$ is formed . Then, using the above-described procedure for forming the set X_{1s}, an attempt is made to supplement the set $X_s(\lambda)$ with vertices from the remainder E_{os}. As a result of these actions, an intra-stable set $X_s(\lambda)$ will be formed, which is very close in content to the intra-stable set X_s. The degree of difference is regulated by the control parameter λ, which is the threshold value of the size of the neighborhood. The vertices removed from X_s are entered at the end of the remainder E_{os}. Note that if after removing the vertices from the subset X_s it was not supplemented, the subset $X_{1s}(\lambda)$ is excluded from consideration.

7 Organization of Search Procedures Based on Modeling of Adaptive Behavior of Bee Colony

The main parameters of the bee colony method are: the number of agents n_b, the maximum number of iterations L, the initial number of intelligence agents n_r, the limitation

of the maximum number of intelligence agents, the threshold value of the size of the neighborhood λ, etc. [2]. At the beginning of the search process, all agents are located in the hive, *ie* outside the search space [2]. In the first iteration (l=1), intelligence agents in the number n_r are randomly placed in the search space [2]. This operation is to generate a set of lists E=$\{E_s \vee s=1,...,n_r\}$, that differ from each other, randomly, which corresponds to the set of items in A=$\{a_s \vee s=1,...,n_r\}$.

For each list E_s, an intra-stable set X_s with the remainder E_{os} is formed, and the value of the objective function F_s is calculated. n_δ basic (best) solutions of $X^\delta = \{X_s\}$ are selected, in which the value of the objective function is not less than the value of the objective function of any unselected solution [2]. A set of basic (best) positions $A^\delta = \{a^{\delta_s} \vee s=1,...,n_\delta\}$ is formed, which correspond to the set of basic (best) solutions X^δ.

Three approaches to determining the number of busy worker bees directed around each neighborhood are proposed. In the first approach, foragers are distributed evenly across the base positions. In the second approach, the foragers are distributed over the base positions in proportion to the value of the objective function of the position. In the third approach, the probabilistic choice is realized [2]. The probability $P(a^{\delta_s})$ of selection by the forager of the base position $a^{\delta_s} \in A^\delta$ is proportional to the value of the objective function $F^{\delta}{}_s$ in this position and is defined as $P(a^{\delta}{}_s) = F^{\delta}{}_s / \sum_s (F^{\delta}{}_s)$ [2].

In the first and second approaches, the number of solutions in the neighborhoods is calculated, and in the third approach - is determined randomly. After the forager b_z selects the base position $a^{\delta}{}_s \in A^\delta$, the probabilistic selection of the position a_z located in the vicinity of the base position $a^{\delta}{}_s$ is realized. The probabilistic choice of the position a_z and the formation of the corresponding solution is carried out in accordance with the procedure described above. In this case, the number of vertices λ_z that are removed and lies within $1 \le \lambda_z \le \lambda$ is pre-randomly determined.

Let us denote the set of positions selected by foragers in the neighborhood of position $a^{\delta}{}_s$ as $O^{\delta}{}_s$. Let's call the set of positions $O^{\delta}{}_s \cup a^{\delta}{}_s$ area $D^{\delta}{}_s$.

In each area $D^{\delta}{}_s$, the best position $a^*{}_s$ is chosen with the best estimate of $F^*{}_s$. Let's call $F^*{}_s$ an estimate of the area $D^{\delta}{}_s$. Among $F^*{}_s$ the best estimation F^* and the corresponding decision, found on the given iteration together by both free and busy worker bees, is chosen. The best solution with a score of F^* is saved, and then there is a transition to the next iteration. Note that in the considered paradigm of the bee colony it is not important to know which agent (worker bee) selected position in the search space. It is important to know the number of free and busy worker bees and also which positions are selected by free worker bees and which are selected by busy worker bees.

In the second and subsequent iterations, the set of base positions $A^\delta(l)$, where (l=2,3,...,L) is formed from two parts $A^{\delta 1}(l)$ and $A^{\delta 2}(l)$, where respectively $A^{\delta 1}(l) \cup A^{\delta 2}(l)$=$A^\delta(l)$. The first part $A^{\delta 1}(l)$ includes $n_{\delta 1}$ best positions a^s found by agents in each of the areas formed in the previous iteration. The second part $A^{\delta 2}(l)$ is formed by scout bees as well as in the first iteration. The difference is in the number n_{r1} of scout agents who randomly select new positions. n_{r1}<n_r. The set $A^{\delta 2}(l)$ includes $n_{\delta 2}$ best

positions from n_{r1} new positions found by scout agents on the l-th iteration. $n_{\delta 1}+n_{\delta 2}=n_{\delta}$. Then actions similar to the actions considered in the first iteration are performed. The number of forage agents that are sent to the neighborhood of each base position is calculated. Each forage agent b_z selects a base position $a_s(l)$ and a position $a_z(l)$ located in the neighborhood of this base position.

In each area $D^{\delta_s}(l)$, the best position $a^s(l)$ with the best solution $F^s(l)$ is selected. The best one $F^*(l)$ is chosen among the estimates $F^*{}_s(l)$. If $F(l)$ is better than $F(l-1)$, then the solution with this estimate is saved, and then there is a transition to the next iteration.

The scheme of operation of the swarm algorithm includes the following steps [1-9]:

1. The main parameters of the bee colony method are set: L is maximum number of iterations; n_r is the initial number of scout agents; n_δ is number of base positions; λ is threshold value of the size of the neighborhood; n_f is the initial number of foragers; $n_{\delta 1}$ is the number of base positions formed from the best $a^*{}_s(l)$ positions found by the swarm on the l-th iteration; n_{rl} is the number of scout agents which randomly select new positions; n_{δ_2} is the number of base positions that are formed from the best new positions that are found by intelligence agents on the l-th iteration;

2. $l = 1$ (l - iteration number);

3. Generate a set of lists $E(l)=\{E_s(l)|s=1,...,n_r\}$ that differ from each other randomly, which corresponds to a set of items $A(l)=\{a_s(l)|s=1,...,n_r\}$.

4. For each list $E_s(l)$, an intra-stable set $X_s(l)$ with a remainder $E_{os}(l)$ is formed and the value of the objective function $F_s(l)$ is calculated;

5. A set of basic solutions $X^\delta(l) \subset X(l)$ with the best values of objective functions $F_s(l)$ and the corresponding set of basic positions $A^\delta(l) \subset A(l)$ are formed. $|A^\delta(l)| = |X^\delta(l)|=n_\delta$;

6. $z = 1$ (z - serial number of the forage agent);

7. Selection with probability $P(a^{\delta_s})=F^{\delta_s}/\sum s(F^{\delta_s})$ of the base position $a^{\delta_s}(l) \in A^\delta(l)$;

8. Probabilistic selection of position $a_z(l)$ located in the neighborhood of the base position $a^{\delta_s}(l)$, with the appropriate decision $X_z(l)$;

9. If the position $a_z(l)$ coincides with the previously selected positions, then go to 8, otherwise go to 10;

10. The position $a_z(l)$ is included in the set $O_s(l)$;

11. Calculation of the value of the objective function $F_z(l)$ of the decision $X_z(l)$;

12. If $z<n_f$, then $z=z+1$ and go to 7, otherwise go to 13;

13. Formation for each base position $a^{\delta_s}(l)$ an area $D_s(l)=O_s(l) \cup a^{\delta_s}(l)$;

14. In each area $D_s(l)$ the best position $a^*{}_s(l)$ with the best decision $X^*{}_s(l)$ is chosens;

15. Among $X^*{}_s(l)$ the best decision $X^*(l)$ is chosen;

16. If $X^*(l)$ is better than $X^*(l-1)$, then it is stored, otherwise $X^*(l)=X^*(l-1)$;

17. If $l<L$, then $l=l+1$ and go to 18, otherwise go to 22;

18. The first part $X^{\delta 1}(l)$ includes $n_{\delta 1}$ preferred positions, among the positions $x^*_s(l-1)$ found by agents in each of the $D_s(l-1)$ areas formed in the previous iteration;

19. Randomly generate sets and lists $E(l)=\{E_s(l)|s=1,...,n_{r1}\}$ that differ from each other, which corresponds to the set of positions $A(l)=\{a_s(l)|s=1,...,n_{r1}\}$, $|E(l)=n|$;

20. Inclusion in the set $A^{\delta 2}(l) n_{\delta 2}$ best positions from the set $A(l)$ of new positions found by scout agents on the l-th iteration $n_{\delta 1}+n_{\delta 2}=n_{\delta}$;

21. Formation of a set of basic positions $A^{\delta}(l)=A^{\delta 1}(l) \cup A^{\delta 2}(l)$. Go to 6;

22. The end of the algorithm. Decision $X^*(l)$ is the best solution found by a swarm of agents.

Temporary complexity of this algorithm depends on the lifetime of the colony l (number of iterations), the number of positions c and the number of agents m and is defined as $O(l * c^2 * m)$.

8 Building Information Model Tourist Site

Nowadays, the Internet has become an indispensable source of information for people planning their vacation, recreation or entertainment. To search for the necessary information on the Internet, users can use, for example:

- Tourist sites;
- Bookmarking services;
- Site directories;
- Recommendations of friends;
- Sites of organizations that provide travel services;
- Search engines (SE), entering the required query.

One of the best ways to get the information you are looking for is to use convenient travel sites. However, there is a noticeable lack of sites with the necessary information and good functionality. A significant part of the Ukrainian web space for tourism is occupied by sites of the two most common types:

- Sites whose main task is to display information about sanatoriums, boarding houses, hotels of resort towns and the functionality of ordering or booking places in these institutions;
- Amateur sites, which mostly provide information about places of interest and articles by authors about their vacation.

The information model is the basis of a modern site, it facilitates further design of the site structure and site creation in particular.

Currently, most of the available information models of sites are general, the use of which in a particular area requires significant improvements. Note that for some types of sites there are well-developed information models, in particular for:

- News sites;
- Blogs;
- Forums;
- Online stores;
- And others.

Construction of a typical information model of a tourist site will provide an opportunity in the future to create convenient sites for presenting a variety of information in the field of tourism, in particular about:

- Attractions;
- Accommodation facilities for tourists;
- Recreational facilities;
- Health facilities;
- Food establishments;
- Entertainment establishments;
- And others.

Most tourism sites are characterized by a lack of information model, which is:

- Clearly understandable for the user;
- Effective in terms of technical support;
- Convenient in the process of adaptation to partial cases.

As a result, the construction of an information model of a tourist site is an urgent task of research in the direction of creating problem-oriented sites.

One of the popular areas of research is the development of methods for creating sites using content management systems such as Joomla, Drupal, Typo3, WordPress and others [1-4]. The use of information technology in tourism is described [5-8]. Problems of data integration of the tourist sphere are investigated in [5-11]. However, the mentioned studies did not analyze the formalization of tourist sites and the construction of information models in particular [12-21]. The analysis of the mentioned researches shows that at present there is a gap concerning typical information models of tourist sites. The aim is to build an information model of a tourist site that would meet the following requirements [22-35]:

- was clearly understandable to the user;
- was effective in terms of technical support;
- allowed to store a variety of information about different types of tourist objects (tourist object can be a building, a mountain and a city or a certain area);
- provided the ability to adapt to create partial cases of travel sites, for example:
    - Things to do in Ukraine;
    - Kiev entertainment website;
    - site of Western Ukraine museums;
- provided the opportunity to create travel sites based on it with the following functions for users:

- view detailed information about the tourist object, its properties and comments to it;
- view the list of tourist attractions:
  - by category;
  - compared to selected tourist object:
    - location relative to the selected tourist object not further than the specified distance (Fig 1. shows the location of some tourist sites, and we notice that at a distance not greater than R_1 relative to B_1, Γ_2 and P_2 are located, and at a distance not greater than R_2 relative to B_1, P_1, P_2, Γ_2, Γ_3 and B_2 are located);
    - territorially belong to the selected tourist object;
    - recommended by the selected tourist attraction;
    - whose popularity depends on the chosen tourist attraction;
  - in relation to the selected tourist object by categories (Fig. 1 also shows that you can choose, for example, hotels that are at a distance of not more than R_2 relative to B_1, it will be - Γ_2 and Γ_3);
- view materials to tourist sites (articles, news, photos, events, etc.) and comments to them;
- view tourist routes and comments to them;

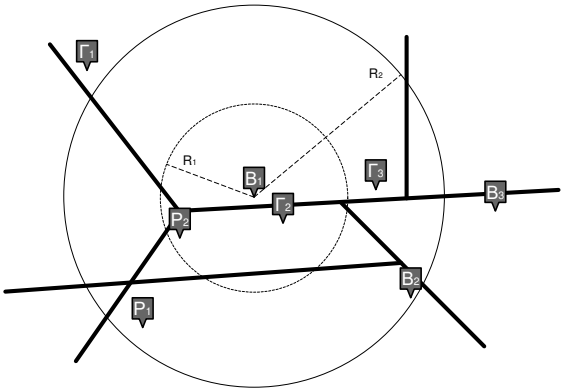

Fig. 1. Example of schematic placement of tourist objects where: R_1, R_2 - distances; markers indicate tourist attractions, the letters in the middle of the markers indicate the category: B - landmark, Γ – hotel, P - restaurant.

The article considers the basic information model, which is built in relation to the basic task of the tourist site, namely: display of information about tourist attractions. However, a modern travel site may have additional functional modules, such as:

- user subsystem:
  - delimitation of user rights, including the allocation of the role of moderator;
  - addition of tourist facilities, materials and routes by users;

- rating subsystem:
    - rating of tourist attractions;
    - rating of materials;
    - user rating;
- module of basic elements of the social network;
- module focused on site positioning in Word Wide Web;
- subsystem of integration with web services and other tourist sites;
- subsystem for searching information on the site;
- subsystem for booking rooms in hotels and other establishments;
- tourist tour booking module;
- other specialized modules.

Due to additional requirements to the functionality of the site, it may be necessary to expand the information model. The main element of the tourist site is the Tourist object (TO) - an object of the tourist sphere, which is potentially interesting for tourists. We will describe what information needs to be stored in accordance with the requirements specified above:

- information about TO:
    - basic properties (name, description, category of TO, the geographical coordinates). *Storing geographical coordinates will allow to calculate the distances between objects; select objects that are close to a specific object; display TO location on the map;*
    - additional set of properties. Which properties will be included in the additional set should depend on the category of TO (the implementation of this feature is discussed below in example 2);
- directory of TO categories with the ability to define subcategories. Examples of categories:
    - regions;
        - settlements;
    - attractions;
        - locks;
        - monuments;
        - ...;
    - accommodation facilities: hotels; sanatoriums; ...;
- relationship between TO;
- materials for TO, divided by types (articles, news, photos, events, etc.);
- user comments on TO (*comments to the TO should characterize the TO, and not, for example, the quality of the text description or design of the TO on the site*);
- user comments on the materials (comments on the materials should characterize the quality of the material);

- routes indicating the tourist attractions through which a particular route passes.

Example 1. Let's talk about the city of Truskavets, which includes the following TO: water "NAFTUSIA"; *Sanatorium "A*"; *Restaurant "B"*; *Spa-Center "C". Restaurant "B"* is located in close proximity to the *sanatorium* "A". We assume that:

- Prosperity of sanatorium *"A"* depends on the availability of healing water "NAFTUSIA" in the city;
- The success of the restaurant "B" depends on the location near the *sanatorium "A";*

It is also assumed that *sanatorium "A"* recommends *restaurant "B"* and *SPA-center "C"*. Fig.2. shows a diagram of the relationships described in this example between TO.

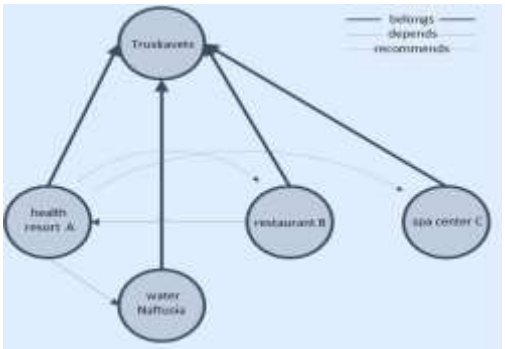

Fig. 2. The scheme of interrelations between tourist objects.

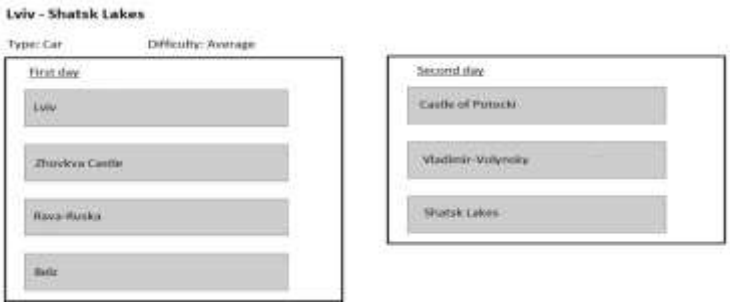

Fig. 3. Example of presenting a tourist route

9 The Entities and Connections Between Them

The entities described in this section and the relationships between them are presented in the ER diagram (Fig. 4). The information model of the tourist site contains the following entities:

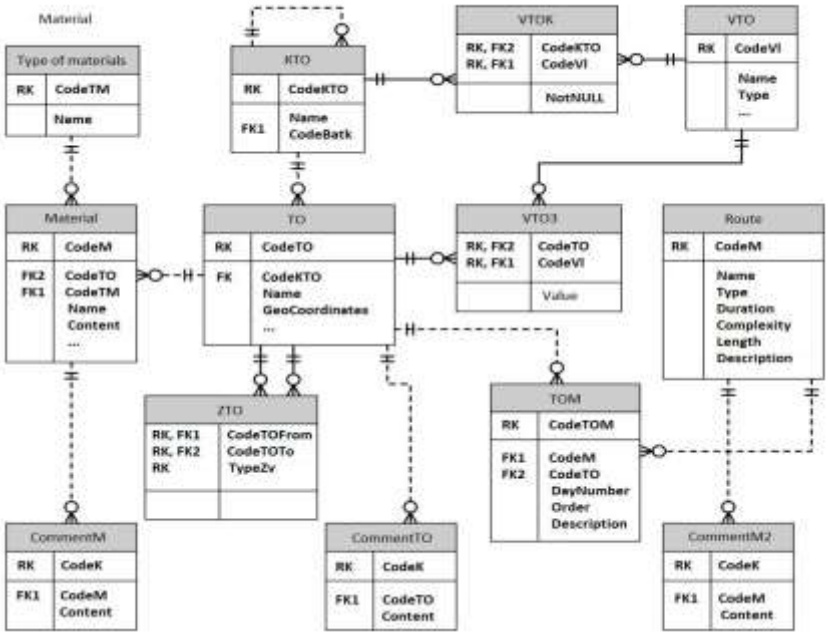

Fig. 4. ER-diagram of the information model of the tourist site

Tourist object (TO) - contains basic information about the tourist object and has the following attributes: *CodeTO*, *CodeTOC* (code of category of TO), *Title*, *GeoCoordinates, Description*. Tourist facilities are interconnected as many-to-many, in particular, we highlight the following types of connections:

1) Territorial affiliation (many-to-many);
2) Dependence (many-to-many);
3) Recommendation (many-to-many).

To implement these connections, the entity of **Tourist Object Relationships (TOR)** is introduced.

*** Tourist Object Relationships -** contains information about connections between tourist objects. This entity has the following attributes: *CodeTOFrom* (the code of the TO-initiator), *CodeTOTo* (code of target TO), *RelType* (type of relationships, can take the values 1, 2, 3).

*** Tourist Object Category (TOC)** - contains information about the categories of tourist attractions. The entity of **Tourist Object Category** has a hierarchical structure. This entity has the following attributes *CodeTOC*, *Title*, *CodeParC* (parent category code, 0 - the top level of the hierarchy).

Relationships between **Tourist Object Category - Tourist Object** are one-to-many.

We present the subsystem of tourist facilities from example 1 in tabular form using the constructed information model as follows:

Table 1. Fragment of the data of Tourist Object Category entity

CodeTOC	Title	CodeParC
…	…	**…**
2	Cities	**0**
…	…	**…**
4	Sanatoriums	**0**
…	…	**…**
7	Restaurants	**0**
…	…	**…**
11	Vacation	**0**
…	…	**…**
19	Mineral waters	**0**
…	**…**	**…**

Table 2. Fragment of the data of the Tourist Object entity.

CodeTO	CodeTOC	Title	…
…	…	…	**…**
1	2	Truskavets	**…**
2	4	A	**…**
3	7	B	**…**
4	11	C	**…**
5	19	Naftusia	**…**
…	**…**	**…**	**…**

Table 3. Fragment of the data of the Tourist Object Relationships entity

CodeTOFrom	CodeTOTo	RelType
…	…	**…**
2	1	**1**
3	1	**1**
4	1	**1**
5	1	**1**
2	5	**2**
3	2	**2**
2	3	**3**
2	4	**3**
…	**…**	**…**

Since the object of each category may have its own set of properties, n categories would require n entities with tourist objects, however, in this implementation, this problem is solved with the help of three additional entities, namely:

*** Tourist Object Attribute (TOA)** – directory of the properties of tourist attractions. The **Tourist Object Attribute** entity has the following attributes: *CodeAt*, *Title*, *Type* (property data type: text, numeric, etc.).

The **TOAC (Tourist Object Attribute Category)** entity is used to implement the M: N connection between the **TOC** and **TOA** entities. *** TOAC** – contains information about what properties must be specified for the TO of a particular category. TOAC has attributes: *CodeTOC* (code of TO category), *CodeAt* (code of TO attribute), *NotEmpty* (indicates whether this property of the tourist object must not take an empty value).

The **TOAR (Tourist Object Attribute Relationships)** entity is used to realize the many-to-many connection between the entities of **TO** and **TOA. * TOAR** – retains the value of additional TO properties. This entity has the following attributes: *CodeTO*, *CodeAt* (code of TO attribute), *Value*.

Example 2. Suppose we have a category "Hotels", and all hotels must contain the parameter "Number of stars", we will provide such information according to the constructed information model as follows:

Table 4. Fragment of the data of Tourist Object Category entity

CodeTOC	Title	CodeParC
…	…	**…**
3	Hotels	**0**
…	**…**	**…**

Table 5. Fragment of the data of Tourist Object Attribute entity

CodeAt	Title	Type
…	…	**…**
37	Number of stars	**integer**
…	**…**	**…**

Table 6. Fragment of the data of Tourist Object Attribute Category entity

CodeTOC	CodeAt	NotEmpty
…	…	**…**
3	37	**1**
…	**…**	**…**

Table 6 shows that for all tourist sites with CodeTOC of 3 it is necessary to enter the CodeAt of 37.

Table 7. Fragment of the data of the Tourist Object entity

CodeTO	CodeTOC	Title	…
…	…	…	**…**
45	3	Hotel "Lviv"	**…**
…	**…**	**…**	**…**

Table 8. Fragment of the data of the Tourist Object Attribute Relationships entity

CodeTO	CodeAt	Value
…	…	**…**
45	37	**4**
…	**…**	**…**

From the information given in tables (4-8) it follows that Hotel "Lviv" is a four-star hotel. One of the advantages of this model is that when adding a new property during operation of the system it is not necessary to change the data structure.

*** Material -** contains information about materials of the TO. This entity has the following attributes: *CodeM*, *CodeTM* (code of type of material), *CodeTO*, *Title*, *Content* (text of material).

*** Type of Materials** - contains information about types of materials. This entity has the following atributes: *CodeTM*, *Title*.

The relationship between **Type of Materials** and **Material** is one-to-many.

The relationship between **TO** and **Material** is one-to-many.

*** CommentM**- contains information about comments on materials. This entity has attributes: *CodeC*, *CodeM* (code of material), *Content* (text of comment).

The relationship between **Material** and **CommentM** is one-to-many.

*** CommentTO** - contains information about comments on TO. This entity has attributes: *CodeC*, *CodeM* (code of material), *Content*.

The relationship between **TO** and **CommentTO** is one-to-many.

*** Route -** contains information about routes. This entity has the following attributes: *CodeR (route code)*, *Title*, *Type* (pedestrian, bicycle, car, public transport or other), *Duration* (number of days), *Complexity*, *Length*, *Description* (general description of the route).

*** TORP (Tourist Object Route Points) -** contains information about intermediate points (tourist sites) of the route. This entity has the following attributes: *CodeTORP*, *CodeR* (route code), *CodeTO*, *DayNumber, Sequence, and Description*. It is worth noting that in one route the same TO can be included more than once.

The relationship between **Route** and **TORP** is one-to-many.

The relationship between **TORP** and **TO** is one-to-many.

*** CommentM2** - contains information about comments to routes, has the following attributes: *CodeC*, *CodeR*, *Content*.

The relationship between **Route** and **CommentM2** is one-to-many.

Let us present the route from Figure 3 in accordance with the constructed information model as follows:

Table 9. Fragment of the data of the Tourist Object entity

CodeTO	…	Title	…
…	…	…	…
1	…	Lviv	…
2	…	Rava-Ruska	…
3	…	Belz	…
4	…	Volodymyr-Volynskyi	…
…	…	…	…
11	…	Zhovkva Castle	…
12	…	Potocki Palace	…
…	…	…	…
21	…	Shatsk Lakes	…
…	…	…	…

Table 10. Fragment of the data of the Route entity

CodeR	Title	Duration	Type	Complexity	…
…	…				…
1	Lviv – Shatsk Lakes	2	car	Medium	…
…	…				…

Table 11. Fragment of the data of the TORP entity

CodeTORP	CodeR	CodeTO	DayNumber	Sequence	…
…	…	…			…
…	1	1	1	1	…

…	1	11	1	2	…
…	1	2	1	3	…
…	1	3	1	4	…
…	1	12	2	5	…
…	1	4	2	6	…
…	**1**	**21**	**2**	**7**	…

One of the effective ways to obtain the desired tourist information is to use specialized travel sites with good functionality, however, the number of such sites in the web space is insufficient. The information model is the basis of a modern site. High-quality information model makes it possible to facilitate the design and creation of the site. Most of the available information models of sites are difficult to adapt to the needs of the travel site without significant improvements. The constructed information model of the tourist site meets certain requirements and allows to develop the tourist site for the purpose of qualitative satisfaction of needs of the user. Potential opportunities for presenting information in the field of tourism with the help of the constructed information model are demonstrated.

10 Conclusions

The article considers a modified paradigm of bee colony for tourist routes by solving combinatorial problems on graphs: selection in the graph of an independent subset of vertices, finding the maximum pairwise combination in the graph, coloring of the graph, selection of a clique of a graph. Based on the analysis of the behavioral model of self-organization of a bee colony, methods and mechanisms for the formation of appropriate representations of solutions of the considered combinatorial problems on graphs are developed. Methods of forming search space are considered. The position in the search space is represented as an ordered list. The key operation of the bee algorithm is the study of promising positions and their neighborhood in the search space. Based on the different applications of the bee colony method, the following advantages of the method can identified:

1. the method is prone to looping in the local optimum because it is based on a random search;
2. multi-agency implementation;
3. the search for a better solution is based on the decisions of agents from the entire bee colony;
4. can used in dynamic applications because it is able to adapt to changes in the environment;
5. can used to solve both discrete and continuous optimization problems;

In addition, the article proposes a method of forming decision environments with an adjustable degree of similarity and closeness between them. It is proposed to use three approaches to determine the number of forage agents sent to the neighborhood of each base position.

References

1. Weise, T.: Global Optimization Algorithms – Theory and Application: Ph.D. thesis, University of Kassel. (2008)
2. Lebedev, B.K., Lebedev, O.B., Lebedeva, E.M., Kostyuk, A.I.: Integration of Models of Adaptive Behavior of Ant and Bee Colony. In: Advances in Intelligent Systems and Computing book series, AISC, 764, 174-185. (2018)
3. Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.:The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK (2005).
4. Basturk, B., Karaboga D.: An artificial bee colony (abc) algorithm for numeric function optimization. In: IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, (2006)
5. Subbotin, S. A., Oleinik, Al. A.: Multiagent optimization based on the bee-colony method. In: Cybernetics and Systems Analysis volume 45, pages177–186(2009)
6. Leonov, A. V.: Application of bee colony algorithm for FANET routing. In: 2016 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM) (2016)
7. Shakhovska, N., Shakhovska, K., Fedushko, S.: Some Aspects of the Method for Tourist Route Creation. In: Advances in Artificial Systems for Medicine and Education II, 902, 527-537. (2019)
8. Antonyuk, N., Medykovskyy, M., Chyrun, L., Dverii, M., Oborska, O., Krylyshyn, M., Vysotsky, A., Tsiura, N., Naum, O.: Online Tourism System Development for Searching and Planning Trips with User's Requirements. In: Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, 1080, 831-863. (2020)
9. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK (2005).
10. Antonyuk, N., Vysotsky, A., Vysotska, V., Lytvyn, V., Burov, Y., Demchuk, A., Lyudkevych, I., Chyrun, L., Chyrun, S., Bobyk, I.: Consolidated Information Web Resource for Online Tourism Based on Data Integration and Geolocation. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, 15-20. (2019)
11. Vysotsky, A., Lytvyn, V., Vysotska, V., Dosyn, D., Lyudkevych, I., Antonyuk, N., Naum, O., Vysotskyi, A., Chyrun, L., Slyusarchuk, O.: Online Tourism System for Proposals Formation to User Based on Data Integration from Various Sources. In: International Conference on Computer Sciences and Information Technologies, CSIT, 92-97. (2019)
12. Lytvyn, V., Vysotska, V., Burov, Y., Demchuk, A.: Architectural ontology designed for intellectual analysis of e-tourism resources. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 335-338. (2018)
13. Lozynska, O., Savchuk, V., Pasichnyk, V.: Individual Sign Translator Component of Tourist Information System. In: Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, Springer, Cham, 1080, 593-601. (2020)
14. Savchuk, V., Lozynska, O., Pasichnyk, V.: Architecture of the Subsystem of the Tourist Profile Formation. In: Advances in Intelligent Systems and Computing, 561-570. (2019)
15. Zhezhnych, P., Markiv, O.: Recognition of tourism documentation fragments from webpage posts. In: 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET, 948-951. (2018)
16. Zhezhnych, P., Markiv, O.: Linguistic comparison quality evaluation of web-site content with tourism documentation objects. In: Advances in Intelligent Systems and Computing 689, 656-667. (2018)

17. Zhezhnych, P., Markiv, O.: A linguistic method of web-site content comparison with tourism documentation objects. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 340-343. (2017)
18. Babichev, S.: An Evaluation of the Information Technology of Gene Expression Profiles Processing Stability for Different Levels of Noise Components. In: Data, 3 (4), 48. (2018)
19. Babichev, S., Durnyak, B., Pikh, I., Senkivskyy, V.: An Evaluation of the Objective Clustering Inductive Technology Effectiveness Implemented Using Density-Based and Agglomerative Hierarchical Clustering Algorithms. In: Advances in Intelligent Systems and Computing, 1020, 532-553. (2020)
20. Lytvyn, V.V.: An approach to intelligent agent construction for determining the group of bank risk basing on ontology. In: Actual Problems of Economics (7), 314-320. (2011)
21. Dosyn, D., Lytvyn, V., Kovalevych, V., Oborska, O., Holoshchuk, R.: Knowledge discovery as planning development in knowledgebase framework. In: Modern Problems of Radio Engineering, Telecommunications and Computer Science, TCSET, 449-451. (2016)
22. Lypak, O.H., Lytvyn, V., Lozynska, O., (...), Rzheuskyi, A., Dosyn, D.: Formation of Efficient Pipeline Operation Procedures Based on Ontological Approach. In: Advances in Intelligent Systems and Computing, 871, 571-581. (2019)
23. Peleshchak, R., Lytvyn, V., Peleshchak, I., Olyvko, R., Korniak, J.: Decision making model based on neural network with diagonalized synaptic connections. In: Advances in Intelligent Systems and Computing, 853, 321-329. (2019)
24. Pasichnyk, V., Lytvyn, V., Kunanets, N., (...), Bolyubash, Y., Rzheuskyi, A.: Ontological approach in the formation of effective pipeline operation procedures. In: 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 80-83. (2018)
25. Lytvyn, V., Uhryn, D., Fityo, A.: Modeling of territorial community formation as a graph partitioning problem. In: Eastern-European Journal of Enterprise Technologies, 1(4), 47-52. (2016)
26. Lytvyn, V.: The similarity metric of scientific papers summaries on the basis of adaptive ontologies. In: Proceedings of 7th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH, 162. (2011)
27. Lytvyn, V.V., Tsmots, O.I.: The process of managerial decision making support within the early warning system. In: Actual Problems of Economics, 149(11), 222-229. (2013)
28. Kravets, P.: The control agent with fuzzy logic. In: Perspective Technologies and Methods in MEMS Design, MEMSTECH, 40-41. (2010)
29. Kravets, P.: The game method for orthonormal systems construction. In: The Experience of Designing and Application of CAD Systems in Microelectronics, , 296-298. (2007)
30. Kravets, P., Kyrkalo, R.: Fuzzy logic controller for embedded systems. In: International Conference on Perspective Technologies and Methods in MEMS Design, , 58-59. (2009)
31. Berko, A., Alieksieiev, V.: A Method to Solve Uncertainty Problem for Big Data Sources. In: International Conference on Data Stream Mining and Processing, DSMP, 32-37. (2018)
32. Berko, A.Y., Aliekseyeva, K.A.: Quality evaluation of information resources in web-projects. In: Actual Problems of Economics, 136(10), 226-234. (2012)
33. Berko, A.Y.: Models of data integration in open information systems. In: Actual Problems of Economics, (10), 147-152. (2010)
34. Berko, A.Y.: Methods and models of data integration in E-business systems. In: Actual Problems of Economics (10), 17-24. (2008)
35. Berko, A.: Consolidated data models for electronic business systems. In: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 341-342. (2007)