XLVII Symposium on Operational Research
SYM-OP-IS 2020
Belgrade, 20th-23rd September 2020.
UNIVERSITY OF BELGRADE
FACULTY OF TRANSPORT AND TRAFFIC ENGINEERING

BEE COLONY OPTIMIZATION FOR DECISION MAKING PROBLEMS

TATJANA DAVIDOVIĆ[1]

[1] Mathematical Institute of the Serbian Academy of Science and Arts, Kneza Mihaila 36, 11000 Belgrade, Serbia, tanjad@mi.sanu.ac.rs

---

Abstract: *Artificial Intelligence (AI) is a modern field in computer science that involves automatic learning processes. It usually means that computers are programmed to make some decisions based on the input parameter values as well as on the previous experience. Decision making problems were usually not considered as optimization problems, however, recent trends in merging optimization and AI changed that perspective. Metaheuristics were successfully applied to many decision making problems, such as training of Artificial Neural Networks, medical therapy determination, clustering, feature selection, and satisfiability problems. Several recent papers, considering the application of Bee Colony Optimization (BCO) to the decision making problems, are reviewed here with an emphasis on the author's results.*

Keywords: *Artificial intelligence, Decision support systems, Optimization problems, Metaheuristics*

1. INTRODUCTION

Many decision making problems can be formulated as packing, scheduling, clustering, location, etc. These are the well-known optimization problems that can be described as follows. Let $f : S \to \mathbb{R}$ be a real function defined on the domain S and let $X \subseteq S$ be a given set. An optimization problem is to find $\min f(x)$ under the constraint $x \in X$. $f(x)$ is called *objective function*. Domain S represents *solution space*, while X contains *feasible solutions*, the ones that satisfy given constraint. *Optimal solution* or *global optimum* is any solution $x^* \in X$ such that $f(x^*) \leq f(x)$ for all $x \in X$. All other feasible solutions are called *sub-optimal* or *approximate* ones.

Some of the well-known decision problems that can be considered as optimization ones are:

- Satisfiability problem (SAT, 3-SAT, MAX-SAT Reasoning under uncertainty, etc.). Applied in electronic circuit design, hardware and software verification, fault detection, product manufacturing, air-traffic control, cryptanalysis, bio-informatics, etc.
- Clustering - grouping object into similar classes. Applications: medicine, marketing, traffic control, document analysis, etc.
- Location - determining the most appropriate places for some facilities. Applications: Emergency services (ambulances, police offices, fire stations), hypermarkets, shopping malls, etc.
- Scheduling - matching two or more entities in space and time. Applications: Multiprocessor systems, aircraft crews, medical staff, production lines, etc.
- Routing: ships along the coast, buses on lines, etc.
- Allocation of resources to clients: clouds, berths in port to the incoming ships, power to machines, workers to tasks, etc.

The usual optimization methods can be classified as *exact*, *heuristic*, *approximation*, or *metaheuristic*. *Exact methods* are the ones that perform exhaustive search over X and find the optimal solution or prove that such a solution does not exist. Having in mind that the optimization problems are usually NP-hard, the application of exact methods is impractical. *Heuristic methods* are used to obtain sub-optimal solutions very fast. They are usually constructive procedures that apply *a priori* knowledge about the considered problem. In most of the cases, the quality of the obtained solution is impossible to estimate. However, for some problems there exists the class of *approximation algorithms*, the heuristic methods with guarantied worst case relation between optimal solution and the estimation proposed by the algorithm. *Metaheuristics* [30] are optimization frameworks (general methods, recipes) to efficiently deal with NP-hard optimization problems. They are incomplete search methods, however, usually provide high-quality solutions within reasonable execution time. Metaheuristics represent iterative, stochastic algorithms that build or improve solutions until the stopping criterion is satisfied. One can distinguish between single-solution or population-based metaheuristics as well as between mathematically founded or nature-inspired methods.

Typical examples of metaheuristic methods are Simulated Annealing (SA), Genetic Algorithms (GA), Tabu Search (TS), Variable Neighborhood Search (VNS), Scatter Search (SS), Greedy Randomized Adaptive Search Procedures (GRASP), Differential Evolution (DE), Estimation of Distribution Algorithms (EDA), Ant Colony

Optimization (ACO), Artificial Bee Colony (ABC), Artificial Immune Systems (AIS), Bee Colony Optimization (BCO), Cooperative Coevolutionary Methods (CCM), Particle Swarm Optimization (PSO), and Memetic Algorithms (MA), known as hybrids between single-solution and population-based methods. Hybridization of two solution methods is very popular [3], especially for particulary hard problems, or in the case when solution space is extremely large, or when the objective function cannot be expressed in the analytical form.

The main goal of this paper is to review the application of BCO to decision making problems. BCO is a Swarm Intelligence (SI) metaheuristic method that was proposed by Lučić and Teodorović [24] and later developed by many researchers all over the world [7]. There are two main variants of BCO, constructive BCO (BCOc) and improvement-based BCO (BCOi) [7].

The paper is organized as follows. After the introduction, some of the well-known decision problems are presented in Section 2. A brief survey of papers considering the application of BCO to these problems and their specific variants is given in Section 3. The paper concludes with Section 4.

2. DECISION MAKING PROBLEMS

Here, the focus is to decision problems that can be formulated as satisfiability, clustering, location, and other well-known NP-hard optimization problems. Some of them are presented in the reminder of this section.

Satisfiability problem. Boolean satisfiability in the propositional logic (referred to as SAT problem) is one of the oldest decision problems. On the other hand, its significance and applicability ranges from integrated circuit design or noise prediction in telecommunication networks to fault diagnosis and safety analysis in modern systems of various kinds [25]. For a given propositional formula φ with n Boolean variables, SAT consists of determining if there exists an valuation (assignment of *true* or *false* value to the variables) such that the formula φ has value *true*. If such a valuation exists, the formula φ is called satisfiable. Otherwise, i.e., if the formula φ has value *false* for all possible valuations, it is unsatisfiable. Formula φ is usually given in the Conjunctive Normal Form (CNF) which means that it is represented by a conjunction of many disjunctions (called *clauses*). In order to prove satisfiability of the formula φ, one should prove that all clauses have value *true*. On the other hand, the clauses being disjunctions are *true* if at least one literal (Boolean variable or its negation) evaluates as *true*. There are various version of SAT problem, the most famous being 3-SAT and MAX-SAT. SAT is known as NP-hard optimization problem [10]. Therefore, exact solvers have to examine 2^n (n being the number of propositional variables) possible valuations to prove that formula is not satisfiable. For checking the satisfiability, the so called incomplete SAT solvers are widely used in the literature [17] and metaheuristics represent significant class of such solvers. ACO for SAT problems was proposed in [32], while in [4], TS, SS, GA and MA for MAX-SAT are studied. In [27] GA for the Probabilistic Satisfiability Problem was developed, while in [21] VNS was proposed for the same problem.

Clustering problems. One of the most important tasks in decision making process reduces to clustering problem [18]. It consists of grouping a set of m objects into K groups called *clusters*. Usually, K is a given input parameter, however, sometimes it is also necessary to determine the most appropriate value for K. The goal of clustering is to group together similar objects while at the same time different object are belonging to distinct groups. Objects are represented as points in n-dimensional space with coordinates called attributes. These attributes are defining various properties of each object. More precisely, an object o^j, $j = 1, 2, \ldots, m$ is represented by $o^j = (a_1, a_2, \ldots, a_n)$. Having in mind that clustering problem is known to be NP-hard [2], heuristic approaches dominate in the relevant literature. From the recent literature ([20]), it can be concluded that the nature-inspired methods (various evolutionary algorithms, GA, DE, ACO, PSO, ABC, AIS) are predominantly used for data clustering. Methods based on local search, such as primal-dual VNS [16] and TS [29], are also applied. An important feature of clustering is the definition of distance between objects [33]. The role of distance is to measure their similarity (or dissimilarity). The distances-based clustering methods are very popular due to their simplicity and applicability in many different cases. The special class of clustering problems include handling of incomplete data. It is a very common case in real application that some values of the attributes describing objects to be clustered are missing [12]. Most of the existing clustering algorithms assume that the objects to be classified are completely described. Therefore, prior to their application the data should be preprocessed in order to determine the values of missing data. Recently, some classification algorithms that do not require to resolve the problem of missing data were proposed. They include Rough Set Models [35], Bayesian Models [22] and Classifier Ensemble Models [34].

Feature selection. This problem consists of identifying and selecting a subset of relevant attributes (features), reducing the dimensionality of data, and therefore, making data processing more efficient [11, 15]. Feature selection problem can be described as follows: for a given original set A of n attributes, find a subset F consisting of p the most relevant features where $F \subset A$ and $p \ll n$. Relevancy for the subset of feature is defined by its influence to the classification accuracy: the goal is to minimize p and losses in accuracy at the same time. Proper solution to the feature selection problem enables discarding remaining $(n-p)$ features and reducing the cost of future data processing with respect to time and space (storage) requirements. The complexity of feature selection problem is similar to the complexity of SAT: an exact selector should evaluate all subsets of features (2^n) and identify the best one. Heuristic approaches to feature selection problem usually involve computing the relative contribution of each attribute to the corresponding processing result. In addition, the correlation index between pairs of attributes is computed. Then attributes are ranked based on the relevance and non-correlated the most influential p of them are selected. Although Artificial Neural Network (ANN) is the most frequently used approach, in the recent literature, metaheuristics such as ABC, ACO, DE, GA, PSO can also be found [23].

The most remarkable example of combining metaheuristics and decision making is certainly the application of metaheuristics to ANN training. In this problem, the goal is to find the most suitable values for weights on synapses. To apply metaheuristics, weights are represented as a vector w and objective function is defined as the Root Mean Square Error (RMSE) measured between the labels from the training set elements and the outputs produced by ANN for a given w. Results obtained by the application of SA, TS, VNS, GRASP, EDA, GA, ACO, CCM, and MA are summarized in book [1]. An interesting example of decision support system designed to help researchers in graph theory is AutoGraphiX (AGX) [5]. The main purpose of AGX is to search for extremal graphs, i.e., graphs minimizing or maximizing a certain graph invariant (or a function of graph invariants). The properties of the generated extremal graphs are used to derive conjectures that can possibly be proved automatically or by humans (researchers). The basic component of this package is the VNS metaheuristic. Among numerous applications of AGX, an interesting one is described in [6] where spectral graph theory was used for reconstruction of a graph with given spectrum and for testing graph isomorphism. It would be interesting to extend results related to these two topics in the future research.

3. BEE COLONY OPTIMIZATION FOR DECISION MAKING

Dose planning in the treatment of thyroid cancer. One of the first papers applying BCO to decision making problem is [31]. The BCOi metaheuristic combined with Case-Based Reasoning (CBR) is proposed for radioiodine (I-131) dose planning in the treatment of thyroid cancer. Having historical records about previous experience (considered as a kind of memory), CBR uses the premise that similar problems have similar solutions and determines the solution for a new instance of the problem based on its similarity with already solved instances from the memory. Therefore, this problem can be considered as supervised clustering. When a physician makes the decision about the proper therapy, it takes into account various clinical parameters (such as primary diagnosis, the patient's age, the size of tumor, the existence of metastases, etc.). The BCOi proposed in [31] is running in two phases: learning and testing. In the first phase, the weights (importance) of clinical parameters are determined by analyzing patients that already underwent the treatment. During the second phase, the proper dose is calculated for newly arrived patients. Actually, the subset of patients is used in the learning phase, while the remaining ones are used for testing. In the learning phase, BCOi is implemented as described in [31]. Solutions are represented by the values of weights for all clinical parameters. The initial solutions are generated randomly, with the restrictions that all weights must be positive (or zero) and their sum for each bee must be equal to one. The objective function to be minimized is represented by the difference between the doses recommended by an artificial bee and the real dose (prescribed by the physician). Clearly, the goal is to make this difference as small as possible. The solution transformation process consists of reducing or increasing a randomly selected weight by 0.1. Obviously, in each forward pass, a pair of randomly chosen weights has to be changed in order to preserve the condition that the sum of weights must be equal to one. When a stopping condition is met, the learned weights are incorporated in CBR to determine radioiodine (I-131) dose for the new patients. To each new patient, the suggested dose matches the one prescribed to the most similar among all previously treated patients. The approach proposed in [31] is tested on data related to real patients treated in the Department of Nuclear Medicine, Clinical Center Kragujevac, Serbia. Comparison results show large similarity between real doses and the ones suggested by the implemented hybridization of BCOi and CBR.

BCOc for feature selection problem. The first application of BCOc to feature selection problem in classification is proposed in [9]. The solution consists of features selected by the corresponding bee. Actually, it is represented by a binary array of length n with 1 on position i indicating that feature a_i is selected and 0 marking that the corresponding feature should be discarded. In each forward pass, bee adds a pre-specified number of randomly selected features to its current partial solution. Each solution is characterized by the fitness defined as the percentage of correctly classified objects, i.e., the ratio between the number of correctly classified objects and the total number of objects multiplied by 100. Loyalty decision in [9] is determined in the following way. Only the bees that generated solutions with fitness value larger than some threshold r are supposed to be loyal.

After all fitness values are calculated, the average of the highest and the lowest fitness values is calculated and denoted by A. In each backward pass, the value for r is selected randomly from the interval $[A-1,A)$. The recruitment of uncommitted bees is performed in standard way. An iteration of BCOc for feature selection is completed when each bee visits all features and decides which ones are to be selected. The authors of [26] proposed the modification of original BCOc for the feature selection problem. The main goal was to reduce the influence of randomness when deciding on loyalty and recruitment. For that purpose, some heuristic rules were introduced. The modified BCOc, called weighted BCOc (wBCO) uses weights during the backward pass to increase the exploitation rate. Two types of weights are proposed for the loyalty decision: global weights (measuring the popularity of given solutions in the current iteration) and local weights (indicating the potential of selected solution for further search). In addition, during the recruitment phase the so called filtering stepwise process (divided into two filtering steps) is used. The first filtering step is based on the similarity between the solutions, while the second one relies on the solution quality. The authors claim that the proposed approach is suitable for classification and regression type problems.

Clustering of incomplete data by BCOi. Clustering of incomplete data based on the p-median classification model was considered in [8]. To avoid data imputation, a new distance function proposed in [14] was used. It is based on Hamming distance and propositional logic formulae. This distance was incorporated into the BCOi metaheuristic and evaluated on standard UCI machine learning repository. Solutions were represented by sets of K cluster representatives, the so-called *centers*. They were actually real objects attracting the closest non-centers according to the selected distance. To improve the efficiency of the proposed BCOi, a pre-processing phase was performed enabling to transform each solution in the constant number of steps. The first initial population (in the first BCOi iteration) was determined randomly. For the remaining iterations, only a half of the population was determined randomly, while $B/2$ population members were assigned the global best solution obtained in one of the previous iterations. During the forward pass, the solution transformation was performed by each bee. The transformation was defined by replacing current centers with some randomly selected non-center objects. At the beginning of iterations, the closest objects were replacing centers, while towards the end all non-center objects were considered. It was possible to leave center unchanged by selecting 0 as an index of the new non-center object. To determine the value of the objective function, each non-center object had to be assigned to the cluster represented by the nearest center and the sum of distances between the objects and the corresponding centers calculated. At the end of each forward pass, best solution held by bees was used to update the global best solution. The backward pass was implemented in standard way [7]. Experimental evaluation showed that the proposed BCOi on average performed better with respect to clustering accuracy than several methods involving learning phases. The same problem was addressed by VNS in [13] and tested on a real-life problem of clustering patients organizing in the database of autoimmune diseases from the Clinical Center of Serbia.

BCOi for 3-SAT. 3-SAT problem is a special case of SAT where clauses of CNF consist only of 3 literals. In [19] an incomplete 3-SAT solver based on BCOi metaheuristic was developed. The main characteristic of the proposed BCOi is the lack of re-initialization of the solutions at the beginning of each iteration. The initial population was determined only once, at the beginning of the first iteration. Later on, final solutions from the previous iteration become initial solutions to the next one. Solution transformation was defined by flipping values of binary variables. A new parameter NCT was introduced to determine the allowed number of flips in one forward pass. Two types of transformations were explored: Random Walk type and Walk SAT type. The first transformation takes randomly one of the unsatisfied clause and flips one of the three variables. Walk SAT transformation is more sophisticated. For each variable x from an unsatisfied clause, first the $break(x)$ is determined as the number of clauses that depend on the current value of x. Then, a variable with minimal value of $break(x)$ is selected for flipping. The main goal in [19] was to perform fine tuning of all BCOi parameters, and therefore, comparisons were performed only with Random Walk and Walk SAT algorithms (obtained by setting $B=1$ in BCOi). The obtained results show that BCOi is able to improve the performance of Random Walk by several orders of magnitude. On the other hand, basic implementation of more sophisticate Walk SAT rules in the BCOi algorithm provide solutions of similar quality as the stand alone Walk SAT solver.

Satisfiability Problem in Probabilistic Logic: BCOi approach. The satisfiability problem with approximate conditional probabilities (CPSAT-ε) was considered in [28]. This problem can be transformed to a system of linear inequalities over probabilities related to literals (atoms). The objective function was defined as the measure of solution infeasibility, and the goal when solving linear system is to annulate its value to zero in order to obtain an optimal solution. For the number of inequalities L and the number of variables n holds $L \ll n$. If the considered formula can be satisfied, the solution of CPSAT-ε can be represented as an array containing $L+1$ atoms $x=[a_1,a_2,\ldots,a_{L+1}]$, with assigned positive probability values $p=[p_1,p_2,\ldots,p_{L+1}]$. BCOi was proposed as the first metaheuristic approach to CPSAT-ε in [28]. At the beginning of each iteration, the initial population was composed of B random solutions. Each solution was generated by randomly selecting

$5(L+1)$ atoms with assigned initial probability equal to $1/(L+1)$. The corresponding grade of each atom (its contribution to the satisfiability of inequalities) was calculated and $(L+1)$ atoms with highest grade were included in the initial solution. In order to transform the solutions, each bee was allowed to change the subset of selected atoms or the probabilities of the existing atoms (or both). Within an iteration of BCOi, each solution was transformed NC times in the following way. First the new $L/5$ atoms with assigned probabilities $1/(L+1)$ were added to the current solution. After removing $L+1$ atoms with the lowest grade, the probabilities of all atoms were recalculated in such a way that their sum equals 1. As a part of solution transformation process, two heuristics known from the literature (worst unsatisfied projection and greedy giveaway) were applied to reduce the infeasibility of the current solution. Within the first heuristic, the most unsatisfied five rows of the linear system were "improved" by changing the probabilities. This procedure was repeated at most 5 times or as long as the improvement occurs. Within the second heuristic the difference between the worst and the best row was reduced by selecting a pair of atoms and changing their probabilities in such a way that the unsatisfiability of the worst row is improved and (at the same time) the unsatisfiability of the best row slightly decreased. The proposed approach was tested on standard benchmark examples of probabilistic formulae and it provided solutions for all satisfiable ones within very short execution time.

4. CONCLUSION

The application of Bee Colony Optimization (BCO) metaheuristic to decision making problems, such as satisfiability, clustering, and feature selection, was surveyed in this paper. The main goal was to promote this simple but powerful Swarm Intelligence optimization method among wider population of researchers, especially the ones working in Artificial Intelligence field. BCO has proven especially useful in the cases when it is necessary to degrade the solution quality in order to further improve it. Therefore, some promising AI applications may include decision making based on scheduling and graph-modeled problems.

Acknowledgement

This work has been supported by Serbian Ministry of Education, Science and Technological Development through the Serbian Academy of Sciences and Arts, Grant No. F159.

REFERENCES

[1] Alba, E., Martí, R. (2006). Metaheuristic procedures for training neural networks. Operations Research/-Computer Science Interfaces Series, vol. 35. Springer Science & Business Media.

[2] Aloise, D., Deshpande, A., Hansen, P., Popat, P. (2009). NP-hardness of euclidean sum-of-squares clustering. Machine learning, 75(2), 245–248.

[3] Blum, C., Puchinger, J., Raidl, G. R., Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. Applied soft computing, 11(6), 4135-4151.

[4] Boughaci, D., Drias, H. (2005). Efficient and experimental meta-heuristics for MAX-SAT problems. In: International Workshop on Experimental and Efficient Algorithms (501–512). LNCS, 3503, Springer, Santorini Island, Greece

[5] Caporossi, G., Hansen, P. (2000). Variable neighborhood search for extremal graphs: 1 The AutoGraphiX system. Discrete Mathematics, 212(1-2), 29-44.

[6] Caporossi, G., Cvetković, D., Rowlinson, P. (2014). Spectral Reconstruction and Isomorphism of graphs using variable neighbourhood search. Bulletin Académie serbe des sciences et des arts. Classe des sciences mathématiques et naturelles. Sciences mathématiques, 39, 23-38.

[7] Davidović, T., Teodorović, D., Šelmić, M. (2015). Bee colony optimization Part I: The algorithm overview. Yugoslav Journal of Operational Research, 25(1), 33–56.

[8] Davidović, T. and Glišović, N. and Rašković, M. (2018). Bee colony optimization for clustering incomplete data. In: Belim, S. et al. (eds.) Proceedings of the School-Seminar on Optimization Problems and their Applications (OPTA-SCL 2018). vol. 2098. CEUR Workshop Proceedings, Omsk, Russia, 94–108.

[9] Forsati, R., Moayedikia, A., Keikha, A. (2012). A novel approach for feature selection based on the bee colony optimization. International Journal of Computer Applications, 43(8), 13–16.

[10] Garey, M.R., Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completness. New York: W. H. Freeman and Company.

[11] Gasca, E., Sánchez, J.S., Alonso, R. (2006). Eliminating redundancy and irrelevance using a new mlp-based feature selection method. Pattern Recognition, 39(2), 313–315.

[12] Gautam, C., Ravi, V. (2014). Evolving clustering based data imputation. In: IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT 2014), 1763–1769.
[13] Glišović, N., Davidović, T., Rašković, M. (2017). Clustering when missing data by using the variable neighborhood search, (in Serbian). Proc. 44th Symposium on Operations Research, SYM-OP-IS 2017, Zlatibor, Sept. 25-28, 2017, 158-163.
[14] Glišović, N., Rašković, M. (2017). Optimization for classifying the patients using the logic measures for missing data. Scientific Publications of the State University of Novi Pazar Series A: Applied Mathematics, Informatics and mechanics, 9(1), 91–101.
[15] Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3, 1157–1182.
[16] Hansen, P., Brimberg, J., Urošević, D., Mladenović, N. (2009). Solving large p-median clustering problems by primal–dual variable neighborhood search. Data Mining and Knowledge Discovery, 19(3), 351–375.
[17] Hoos, H.H., Stützle, T. (2005). Stochastic local search: Foundations and applications. Amsterdam: Elsevier.
[18] Jain, A.K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651–666.
[19] Jakšić Krüger, T. and Davidović, T. (2016). Analyis of the bee colony optmization method on 3-SAT problem. In: Proc. 43rd Symposium on Operations Research, SYM-OP-IS 2016, Tara, Serbia, 297–231.
[20] Jose-Garcia, A., Gómez-Flores, W. (2016). Automatic clustering using nature-inspired metaheuristics: A survey. Applied Soft Computing, 41, 192–213.
[21] Jovanović, D., Mladenović, N., Ognjanović, Z. (2007). Variable neighborhood search for the probabilistic satisfiability problem. In Doerner, K. F., et al., (Eds.) Metaheuristics - Progress in Complex Systems Optimization (173–188), New York: Springer, 173–188.
[22] Lin, H.-C. Su, C.-T. (2013). A selective bayes classifier with meta-heuristics for incomplete data. Neurocomputing, 106, 95–102.
[23] Liu, W., Wang, J. (2019). A brief survey on nature-inspired metaheuristics for feature selection in classification in this decade. In 16th IEEE International Conference on Networking, Sensing and Control (ICNSC), 424-429.
[24] Lučić, P., Teodorović, D. (2001). Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, 441–445.
[25] Marques-Silva, J. (2008). Practical applications of boolean satisfiability. In: 9th IEEE International Workshop on Discrete Event Systems, Goteborg, Sweden, 74–80.
[26] Moayedikia, A., Jensen, R., Wiil, U.K., Forsati, R. (2015). Weighted bee colony algorithm for discrete optimization problems with application to feature selection. Engineering Applications of Artificial Intelligence, 44, 153–167.
[27] Ognjanović, Z., Midić, U., Kratica, J. (2004). A genetic algorithm for probabilistic SAT problem. In Proc. 7th International Conference on Artificial Intelligence and Soft Computing-ICAISC 2004, Zakopane, Poland, 462–467.
[28] Stojanović, T., Davidović, T., Ognjanović, Z. (2015). Bee colony optimization for the satisfiability problem in probabilistic logic. Applied Soft Computing, 31, 339–347.
[29] Sung, C.S., Jin, H.W. (2000). A tabu-search-based heuristic for clustering. Pattern Recognition, 33(5), 849–858.
[30] Talbi, E.G. (2009). Metaheuristics: From Design to Implementation. Hoboken, New Jersey: John Wiley & Sons, Inc.
[31] Teodorović, D., Šelmić, M., Mijatović-Teodorović, L. (2013). Combining case–based reasoning with bee colony optimization for dose planning in well differentiated thyroid cancer treatment. Expert Systems with Applications, 40(6), 2147–2155.
[32] Villagra, M., Barán, B. (2007). Ant colony optimization with adaptive fitness function for satisfiability testing. In: International Workshop on Logic, Language, Information, and Computation. LNCS, 4576, Springer, Rio de Janeiro, Brazil, 352–361.
[33] Wang, F., Sun, J. (2015). Survey on distance metric learning and dimensionality reduction in data mining. Data Mining and Knowledge Discovery, 29(2), 534–564.
[34] Yan, Y-T., Zhang, Y-P., Zhang, Y-W., Du, X-Q. (2017). A selective neural network ensemble classification for incomplete data. International Journal of Machine Learning and Cybernetics, 8(5), 1513–1524.
[35] Zhang, Q., Xie, Q., Wang, G. (2016). A survey on rough set theory and its applications. CAAI Transactions on Intelligence Technology, 1(4), 323–333.