Experimental evaluation of some approximate algorithms of solving the multidimensional knapsack problem

Birute I. Yuhimenko, Svitlana V. Kornilova, Inna O. Asaulyuk, Piotr Kisała, Saule Luganskaya, et al.

Birute I. Yuhimenko, Svitlana V. Kornilova, Inna O. Asaulyuk, Piotr Kisała, Saule Luganskaya, Indira Shedreyeva, "Experimental evaluation of some approximate algorithms of solving the multidimensional knapsack problem," Proc. SPIE 11045, Optical Fibers and Their Applications 2018, 1104517 (15 March 2019); doi: 10.1117/12.2523175

Event: 18th Conference on Optical Fibers and Their Applications, 2018, Naleczow, Poland

Experimental evaluation of some approximate algorithms of solving multidimensional knapsack problem

Birute I. Yuhimenko*[a], Svitlana V. Kornilova[a], Inna O. Asaulyuk[b], Piotr Kisała[c], Saule Luganskaya[c], Indira Shedreyeva[d]
[a]Odessa National Polytechnic University, Shevchenko Avenue,1, 65044 Odessa, Ukraine;
[b]Vinnytsia State Pedagogical University named after Mykhailo Kotsiubynskyi, Vinnytsia, Ukraine;
[c]Lublin University of Technology, Nadbystrzycka 38A, 20-618 Lublin, Poland;
[d]Taraz State University after M.Kh.Dulaty, Taraz, Kazakhstan

ABSTRACT

The results of an experimental study of obtaining approximate solutions of the multidimensional knapsack problem are presented in the paper. Three methods for obtaining these solutions are considered, based on branch-end-bound algorithms with a one-sided branching. One hundred test tasks of different dimensions have been solved for each method. The estimation of the approximation of the solutions obtained in relation to the best solution is provided, as well as the estimation of the computational complexity.

Keywords: knapsack problem, approximate algorithms, branch-and-bound method, experimental evaluation, estimation of the approximation, estimation of the computational complexity

1. INTRODUCTION

The popularity of discrete optimization problems is beyond doubt. Many applied problems are formed as problems of discrete, and more often, integer optimization. Not a small number of optimization problems of a specific structure are reduced to integer problems in order to use their solution methods. In this regard, the availability of effective and fast algorithms is one of the key issues of research on discrete optimization. A lot of publications are devoted to this, there is a whole scientific direction. However, most of these algorithms belong to the NP-complexity class. It is an open question how they can be reduced to algorithms with less computational complexity – to complexity class P. The definition of structural features of models of applied problems, their classification and selection of effective algorithms for various classes is also an open problem.

Finding an exact solution requires considerable computational resources and time, that is why approximate algorithms are being developed. Modern approximate methods are usually combined and contain elements of various algorithms. Structurally they are two-staged: 1st stage – construction of the initial solution and 2nd – improvement of the initial solution. At the first stage, approaches of heuristic algorithms are widely used. They usually work on various likelihoods, do not contain a clear theoretical justification, but predetermine the procedure for forming a vector of solutions. The second stage is the improvement of the solution, using the result of the first stage. In addition to the original solution, the value of the objective function-record is used. This allows for the organization of the process of sifting out of the unpromising variants and deciding on getting an approximate solution. Such algorithms work quickly, provide the user with good enough and, in some cases, optimal, solution. Indeed, the requirement of a mathematically exact optimal solution is not always justified, since the initial information for the implementation of applied problems does not meet the requirements of mathematical rigor.

Exact methods for solving integer linear programming problems (ILP) have been developing in two directions. The cutting-plane methods proposed by Danzig and implemented by Gomori are based on modification of the methods of linear programming. A lot of different algorithms have been developed [1], aimed at finding ways to construct the correct cutting in order to increase the rate of their convergence. There are application packages [2] that implement the developed algorithms, as well as provide the possibility of their immediate modification in the interactive mode. The rate of their convergence is still within the NP complexity. The combinatorial approach [3], which gave rise to the development of

* e-mail: pm1987pm@gmail.com

Optical Fibers and Their Applications 2018, edited by Ryszard S. Romaniuk,
Waldemar Wójcik, Andrzej Smolarz, Proc. of SPIE Vol. 11045, 1104517
© 2019 SPIE · CCC code: 0277-786X/19/$18 · doi: 10.1117/12.2523175

Experimental evaluation of some approximate algorithms of solving multidimensional knapsack problem

Birute I. Yuhimenko*[a], Svitlana V. Kornilova[a], Inna O. Asaulyuk[b], Piotr Kisała[c],
Saule Luganskaya[c], Indira Shedreyeva[d]
[a]Odessa National Polytechnic University, Shevchenko Avenue,1, 65044 Odessa, Ukraine;
[b]Vinnytsia State Pedagogical University named after Mykhailo Kotsiubynskyi, Vinnytsia, Ukraine;
[c]Lublin University of Technology, Nadbystrzycka 38A, 20-618 Lublin, Poland;
[d]Taraz State University after M.Kh.Dulaty, Taraz, Kazakhstan

ABSTRACT

The results of an experimental study of obtaining approximate solutions of the multidimensional knapsack problem are presented in the paper. Three methods for obtaining these solutions are considered, based on branch-end-bound algorithms with a one-sided branching. One hundred test tasks of different dimensions have been solved for each method. The estimation of the approximation of the solutions obtained in relation to the best solution is provided, as well as the estimation of the computational complexity.

Keywords: knapsack problem, approximate algorithms, branch-and-bound method, experimental evaluation, estimation of the approximation, estimation of the computational complexity

1. INTRODUCTION

The popularity of discrete optimization problems is beyond doubt. Many applied problems are formed as problems of discrete, and more often, integer optimization. Not a small number of optimization problems of a specific structure are reduced to integer problems in order to use their solution methods. In this regard, the availability of effective and fast algorithms is one of the key issues of research on discrete optimization. A lot of publications are devoted to this, there is a whole scientific direction. However, most of these algorithms belong to the NP-complexity class. It is an open question how they can be reduced to algorithms with less computational complexity – to complexity class P. The definition of structural features of models of applied problems, their classification and selection of effective algorithms for various classes is also an open problem.

Finding an exact solution requires considerable computational resources and time, that is why approximate algorithms are being developed. Modern approximate methods are usually combined and contain elements of various algorithms. Structurally they are two-staged: 1st stage – construction of the initial solution and 2nd – improvement of the initial solution. At the first stage, approaches of heuristic algorithms are widely used. They usually work on various likelihoods, do not contain a clear theoretical justification, but predetermine the procedure for forming a vector of solutions. The second stage is the improvement of the solution, using the result of the first stage. In addition to the original solution, the value of the objective function-record is used. This allows for the organization of the process of sifting out of the unpromising variants and deciding on getting an approximate solution. Such algorithms work quickly, provide the user with good enough and, in some cases, optimal, solution. Indeed, the requirement of a mathematically exact optimal solution is not always justified, since the initial information for the implementation of applied problems does not meet the requirements of mathematical rigor.

Exact methods for solving integer linear programming problems (ILP) have been developing in two directions. The cutting-plane methods proposed by Danzig and implemented by Gomori are based on modification of the methods of linear programming. A lot of different algorithms have been developed [1], aimed at finding ways to construct the correct cutting in order to increase the rate of their convergence. There are application packages [2] that implement the developed algorithms, as well as provide the possibility of their immediate modification in the interactive mode. The rate of their convergence is still within the NP complexity. The combinatorial approach [3], which gave rise to the development of

* e-mail: pm1987pm@gmail.com

Optical Fibers and Their Applications 2018, edited by Ryszard S. Romaniuk,
Waldemar Wójcik, Andrzej Smolarz, Proc. of SPIE Vol. 11045, 1104517
© 2019 SPIE · CCC code: 0277-786X/19/$18 · doi: 10.1117/12.2523175

algorithms of branch-and-bound method, is more mobile, easily realized under the conditions of the finite set of variants.

However, the problem is that in case of applying a mass solution there is an exponential increase in the number of iterations with the increase in the number of unknown values. The introduced optimality conditions of a method, based on the concepts of estimating a set of variants, is not strict and sometimes leads to a complete enumeration of variants. Therefore, difficulties arise in proving the optimality of the solution obtained.

Openness and flexibility of the method involves the use of a variety of techniques that modify the algorithm itself. This greatly affects the complexity of solving the problem at various stages of computational procedures. Techniques of the expansion (narrowing) of the set of variants in obtaining estimates are generally recognized. However, the expansion itself can be carried out in different ways. For example, the reduction of the ILP problem to the corresponding problem of linear programming makes it possible to obtain a sufficiently good estimate close to the value of the objective function. The accuracy of the evaluation contributes to the rapid fulfillment of the optimality condition of the solution obtained. However, the complexity of the multiple solution of the LP problem is very large. In [4], experimental data on the complexity of the solution using various expansion methods are presented. The method of obtaining estimates by solving several non-integer one-dimensional knapsack problems gives a less accurate estimate of the set of variants, but obtaining this estimate is much less time-consuming. The increase in the number of iterations in this way is easily compensated by the simplicity of obtaining estimates. The second main point of the branch-and-bound method is the partitioning of the set into subsets, based on the ideas of sequential construction of the solution [5].

The consecutive assignment of the value "1" to the components of the solution vector predetermines the method of partitioning the set. Two methods are well known: frontal (simultaneous) [3] and one-sided branching [8]. The complexity of the computational process also significantly depends on the procedure of selecting the component to be specified at a step of the algorithm [6].

An important factor affecting the speed of convergence is the procedure for selecting a set of variants for branching. Two moments stand out: choosing a subset on the tier of the decision tree and choosing a subset if the variant found does not satisfy the optimality condition. The choice in the first case is obvious – the preference is given to a subset with the best estimate, depending on which type the problem relates to – maximization or minimization. As for the choice of a subset – the top of the decision tree after obtaining a variant with a record value, but not satisfying the optimality condition, the choice can be made in various ways.

Quite a long time ago, in [7], the notion of preference functions predetermining this choice was introduced. The following options were considered: selection by the best estimate, random choice with respect to the value of the estimates, and a randomized choice with respect to the values – the best estimates. The efficiency of choice was estimated by the mathematical expectation of the values of the objective functions of the variants obtained with different methods of selection. Sensitive difference was not observed. In principle, this was confirmed in another experimental study [9]. The number of viewed variants directly depends on how many unpromising subsets will be eliminated in the process of finding the optimal solution. Obviously, this depends on all the procedural moments of the computational process in the work of the branch-and-bound method [10], that is, the closeness of the record value of the objective function to the optimal, the accuracy of the resulting estimates of subsets, the successful choice of the component of the solution vector for concretization, which predetermines the decomposition into subsets, etc. In [10], the concept R of the proximity of the estimate to the value of the objective function is used. All those subsets whose estimates are worse than this record value are subject to sifting. The effectiveness of this approach to sifting out variants should be verified experimentally. All possible combined moments improve (in the sense of the speed of convergence) algorithms of the branch-and-bound method, however, the computational complexity, as a rule, does not allow for the obtaining of the optimal solution at solving practical problems of large dimension, at an acceptable period of time.

Currently, close attention is paid to algorithms, the main idea of which is borrowed from wildlife or physical processes occurring in nature. Such algorithms can include algorithms of the ant colony [11], genetic algorithms [12], annealing [13], greedy algorithms [14], neural networks, etc.

The proposed work continues research in this direction. Combinatorial approximate algorithms are considered, experimental data are given that allow for the estimation of the efficiency of obtaining sufficiently good approximate solutions. A more detailed description of the algorithms is given in [15,18,19].

2. SOLUTION

The object of the study is a linear multidimensional knapsack problem. This choice is not accidental. Many applied problems are reduced to this model. In addition, some optimization problems of a distributive nature contain, in their model, the model of the knapsack problem. So, there is a problem in the formulation of [20,21]

$$Z = \max \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \ , \ i = \overline{1,m} \, ;$$

$$x_j \in \{0,1\} \, , \qquad j = \overline{1,n}.$$

Approximate algorithms of the branch-and-bound method with one-sided branching are investigated experimentally. The idea of algorithms in the first stage of their formation is that the priority sequence of indices of the components of the solution vector is constructed, according to which the values "1" are assigned to its components. Let us denote as P this sequence [22,23,24]

$$P = \{j_1, j_2, \dots, j_k, \dots, j_n\}$$

on a condition that

$$x_{j_k} = \begin{cases} 1, & \text{if} \left(b_i - \sum_{j=1}^{k} a_{ij}\right) \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Algorithms belong to the class of approximate ones; the degree of approximation is not numerically determined. There are experimental data for solving 100 tasks for each dimension and method of determining the priority sequence P. The work has been carried out, as usual in approximate algorithms, in two stages. At the first one, called the direct, all components were initially reset to zero. The value "1" was assigned according to the priority queue. The second stage, dual, involves replacing the "1" components with zeros, acting in the following way.

Method a. The idea of a greedy algorithm is used. The priority of the components of the solution vector is predetermined by the contribution of the corresponding component of the vector of the value to the value of the objective function. In other words, the place of j_k is determined by the decomposition method of the component c_j in the ranked nondecreasing sequence. If, $c_{j_1} \geq c_{j_2} \dots \geq c_{j_k} \geq c_{j_n}$, then indices $j_k (k=\overline{1,n})$ define the sequence P.

Method b. Combined way to determine the priority sequence of assignment of "1" to the components of the solution vector. Since the number of positive components predetermines the value of the objective function, and the possibility of assigning the value "1" is controlled by the constraint system, it is advisable to take into account the magnitude of the discrepancy in the constraints in the formation of the priority queue. With each component, a value p_j determined by formula

$$p_j = c_j \sum_{i=1}^{m} (b_j - a_{ij})$$

The sequence of ranked values p_j, in particular, $p_{j_1} \geq p_{j_2} \dots \geq p_{j_k} \geq p_{j_n}$ generates a sequence P.

Method c. The procedure for forming the priority sequence is inspired by the ideas of the behavior of the ant colony when searching for food. The right way is where these biological beings pass more often. A formal imitation of this behavior – where "1" is more common in the vectors of the solution of one-dimensional knapsack problems. The multidimensional knapsack problem is subdivided into m ($i = \overline{1,m}$) one-dimensional non-integer problems.

There are m tasks in the formulation

$$Z = \max \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \ , \ i = \overline{1,m} \, ;$$

$$0 \leq x_j \leq 1 \, , \qquad j = \overline{1,n},$$

optimal solutions to which are $X_i^* = (x_{i_1}^*, x_{i_2}^*, \dots, x_{i_k}^*, \dots, x_{i_n}^*)$.

The priority of the solution vector component is estimated as

$$p_j = \sum\nolimits_{i=1}^{m} x_{ij}^* \,, \qquad j = \overline{1,n}.$$

The priority sequence P is again determined by ranking the values p_j.

3. RESULTS AND DISCUSSION

An experimental study of the proposed methods for determining the approximate solution has been carried out by solving a series of problems. The dimension $(m \times n)$ was changing as follows: (10x20); (20x50); (30x100); (40x500); (50x1000). The parameters c_j and a_{ij} have been formed as independent random integers uniformly distributed in the intervals [1.10] and [1.20], respectively. The parameters b_i have been determined depending on the values a_{ij}. In this case [25,26]

$$b_i = \frac{\sum_{j=1}^{n} a_{ij}}{3}$$

For any pair $(m \times n)$, 100 tasks have been solved. Solutions for each of the above methods for determining the initial solution (the first stage) and its improvement (the second stage) are obtained. Each task has been solved in all three ways. The following have been determined:

- the value of the objective function z_r $(r = a, b, c)$;
- the best value of the objective function z^* for a particular task;
- the number of coincidences $\mathrm{k_r}$ with z^* for each task $(r = a, b, c)$.

The number of coincidences k_r of the values z_r and z^* for two stages of solving one hundred tasks is given in Table 1.

Table 1 Comparison with the best value of the objective function

Method(stage) m × n	a		b		c	
	1	2	1	2	1	2
10x20*	29	58	39	65	29	68
20x50	13	52	18	60	14	60
30x100	9	34	11	60	13	50
40x500	7	55	36	78	14	48
50x1000	20	46	58	90	22	44

*Comparison has been carried out with the exact solution, obtained using the Balash algorithm

Table 2. The deviation from the best value of the objective function is no more than 2%

Method(stage) m × n	a		b		c	
	1	2	1	2	1	2
10x20*	12	14	14	10	24	26
20x50	23	24	34	29	35	36
30x100	40	59	48	35	40	40
40x500	41	40	60	21	70	46
50x1000	52	49	32	8	51	48

*Comparison has been carried out with the exact solution, obtained using the Balash algorithm

R – the proximity of the value of the objective function to the best value, has been estimated as a percentage. If the deviation from the best value of the objective function is acceptable by no more than 2%, then the number of coincidences in the proximity to the best value is given in Table 2.

An estimate of the degree of approximation of the solution in percentage is given in Table 3.

Table 3. Estimation of the degree of approximation of the solution

Method(stage) / m × n	a		b		c	
	1	2	1	2	1	2
*10x20	41	72	64	75	53	94
20x50	46	76	52	89	49	96
30x100	49	93	59	95	53	90
40x500	48	95	96	99	84	94
50x1000	72	95	90	98	73	92

*Comparison has been carried out with the exact solution, obtained using the Balash algorithm

The summary data of the table show that it is practically possible to use any method of obtaining an operative and sufficiently good solution of the problem. No more than a 2% deviation of the value of the objective function from its extreme value is completely justified, since for practical problems the initial information has no mathematical accuracy [27,28,29].

Estimation of the complexity of calculations. The complexity of the calculations is usually defined by the time needed for obtaining a solution, which depends on the length of the input, i.e. on the number of characters that predetermine the original data [16]. However, this is more acceptable when processing data directly involved in the calculations. In determining the complexity of computations that have an iterative character, as in this case, it is more clearly to consider the complexity that depends on the dimension of the problem. In [17], the complexity of computations is considered as the number of arithmetic operations performed when obtaining the original solution. In this paper, an attempt to determine the complexity of the calculations was performed through the number of operations [30,31,32].

Table 4 gives data on the supposed complexity of each of the methods for obtaining approximate solutions. Since the methods (algorithms) do not have an iterative character, the computational complexity does not give a complete estimate. This is additional information for choosing the approach to constructing a sufficiently good initial solution of the problem [33,34,35].

Table 4 Ordinal complexity of computations

Method of solution	List of main actions	The assumed complexity
a	– ordering of the priority sequence	$O(n^2)$
b	– definition of values p_j – ordering of the priority sequence	$O(n^2 + m)$
c	– solving m one-dimensional knapsack problems – definition of values p_j – ordering of the priority sequence	$O(n^2 + (m+1)n)$

4. CONCLUSIONS

The direct experimental study of the considered methods showed that to obtain an approximate solution any of them can be used and give quite a good approximation to the extreme value of the objective function. However, the "b" method is somewhat distinguished as the best among the other two. If the two-percent proximity to the extremum is acceptable, then the result obtained in the second stage using the "b" method, almost absolutely provides extremum, regardless of the dimension of the problem.

As for the complexity of computations, the number of operations, and by the way, the amount of memory used by the computer, does not cause any fear in any way to get the solution. It is more difficult to work according to the method "c", because the solution of the m one-dimensional problems of the container still takes more time and requires more operations during calculation. In our opinion, the "b" method, which is the second one in the list by complexity of computations, raises trust.

The considered methods can be used in precise algorithms in obtaining the initial record value of the objective function. In algorithms for the branch and bound method with a frontal branching method, they can be used to construct a fragment of a decision tree. It is possible to check the solution variant for optimality by having estimated the highest vertices in it.

Note. The computer implementation of a series of tasks was carried out by a Bachelor of Applied Mathematics Serbin D., a graduate of ONPU.

REFERENCES

[1] Minu M. [Matematicheskoe programmirovanie. Teoriya i algoritmyi.], Nauka, Moscow (1990).

[2] Dyakonov V. P. [MATLAB. Polnyiy samouchitel], DMK Press, Moscow (2012).

[3] Pigl Dzh., Murti K., Suini D., Kerel K., "Algoritmyi dlya resheniya zadachi kommivoyazhera," Ekonomiko-matematicheskie metodyi 1(1), 90-107 (1996).

[4] Tkalenko, O.Yu., "K voprosu ocenki slozhnosti algoritmov metoda vetvej i granic," Tezisy dokladov mezhdunarodnoj nauchno-prakticheskoj konferencii Project, Program, Portfolio, Management 1, 88-92 (2017).

[5] Mihalevich, V.S., "Posledovatelnyie algoritmyi optimizatsii i ih primenenie," Kibernetika 1(2), 45-55, 85-89 (1965).

[6] Yuhimenko, B.I., Kozina, Yu.Yu., "Sravnitelnaya harakteristika algoritmov metoda vetvey i granits dlya resheniya zadach tselochislennogo lineynogo programmirovaniya," Trudyi Odesskogo politehnicheskogo universiteta 2, 1[illegible]-204 (2005).

[7] Liepuonite–Yuhimenko B.I., "Ob effektivnosti funktsiy predpochteniya v algoritmah tselochislennogo lineynogo programmirovaniya," Litovskiy matematicheskiy sbornik. Vilnyus 23(1), 134-140 (1983).

[8] Serbin D.A., "Modifikatsiya metoda vetvey i granits dlya resheniya zadachi tselochislennogo lineynogo programmirovaniya s bulevyimi peremennyimi," Tezisyi dokladov mezhdunarodnoj nauchno-prakticheskoj konferencii Project, Program, Portfolio, Management 1, 143-147 (2016).

[9] Sienko A.A., "K voprosu ob uvelichenii skorosti shodimosti algoritmov metoda vetvej i granic," Materialy vseukrainskoj nauchno-metodicheskoj konferencii Problemy ekonomicheskoj kibernetiki 3, 29-35 (2011).

[10] Sigal, I.H., Ivanova, A.P., [Vvedenie v prikladnoe diskretnoe programmirovanie], Fizmatlit, Moscow (2007).

[11] Shtovba S.D., "Muravinye algoritmy: Teoriya i primenenie," Programmirovanie 4, 1-16 (2005).

[12] Gladkov L.A., Kuryajchik V.M. [Geneticheskie algoritmy], Fizmatlit, Moscow (2006).

[13] Levanova, T.V., Lorens, M.A., "Algoritmy muravinoj kolonii i imitacii otzhiga dlya zadachi o p-mediane," Avtomatika i telemehanika 3, 80-88 (2004).

[14] Dyubin, G.N., Korbut, A.A., "Zhadnye algoritmy dlya resheniya zadachi o rance: povedenie v srednem," Sibirskij zhurnal industr. Matematiki 3(4), 68-92 (1999).

[15] Yuhimenko, B.I., Volkova, N.P., "Priblizhennye algoritmy resheniya zadach o mnogomernom rance," Doslidzhennya v matematici i mehanici 22(30), 104-115 (2017).

[16] Hachiyan, L.G., "Polinomialnij algoritm v linejnom programmirovanii", Dokl. AN SSSR 244(5), 1093-1096 (1979).

[17] Vasiliev V.S, "Analiz slozhnosti algoritmov", <https://pro-prof.com/archives/1660> (2017).

[18] Wydra, M., Kisala, P., Harasim, D., Kacejko, P, "Overhead Transmission Line Sag Estimation Using a Simple Optomechanical System with Chirped Fiber Bragg Gratings. Part 1: Preliminary Measurements," Sensors 18(1), 309 (2018).

[19] Tarczynska, M., Gaweda, K., Dajewski, Z., et al., "Comparison of treatment results of acute and late injuries of the Lisfranc joint," ACTA ORTOPEDICA BRASILEIRA 21(6), 344-346 (2013).

[20] Vyatkin. S. I., Romanyuk. A. N., Gotra. Z. Y., et al. "Offsetting, relations, and blending with perturbation functions", Proc. SPIE 10445, 104452B (2017).

[21] Vyatkin, S.I., Romanyuk, A.N., Pavlov, S.V. et al., "Fast ray casting of function-based surfaces," Przeglad Elektrotechniczny 93(5), 83-86 (2017).

[22] Timchenko, L. I., Pavlov, S. V., Kokryatskaya, N. I., et al., "Bio-inspired approach to multistage image processing," Proc. SPIE 10445, 104453M (2017).

[23] Vyatkin, S. I., Romanyuk, S. A., Pavlov, S. V. et al., "Using lights in a volume-oriented rendering," Proc. SPIE 10445, 104450U (2017).

[24] Romanyuk. O. N., Pavlov. S. V., Melnyk, O. V., Romanyuk, S. O., Smolarz, A., et al., "Method of anti-aliasing with the use of the new pixel model," Proc. SPIE 9816, 981617 (2015).

[25] Romanyuk, S. O., Pavlov, S. V., Melnyk, O. V., "New method to control color intensity for antialiasing," Control and Communications (SIBCON), 2015 International Siberian Conference. (2015).

[26] Kovalenko, A. S., Tymchyk, S. V., Kostyshyn, S.V., Zlepko, S. M., Wójcik, W., et al., "Concept of information technology of monitoring and decision-making support," Proc. SPIE 10445, 104452D (2017).

[27] Zlepko, S.M., Tymchyk, S.V., et al., "An informational model of sportsman's competitive activities," Proc. SPIE 10031, 100312N (2016).

[28] Selivanova, K.G., Avrunin, O.G., Zlepko, S.M., et al., "Quality improvement of diagnosis of the electromyography data based on statistical characteristics of the measured signals," Proc. SPIE 10031, 100312R (2016).

[29] Rotshtein, A.P., Posner, M., Rakytyanska, H.B., "Cause and effect analysis by fuzzy relational equations and a genetic algorithm," Reliability Engineering and System Safety 91(9), 1095-1101 (2006).

[30] Rotshtein, A.P., Rakytyanska, H.B., "Diagnosis problem solving using fuzzy relations," IEEE Transactions on Fuzzy SystemsVolume 16(3), 664-675(2008).

[31] Timchenko, L.I., "A multistage parallel-hierarchic network as a model of a neuronlike computation scheme," Cybernetics and Systems Analysis 36(2), 251-267(2000).

[32] Timchenko, L.I., Kutaev, Y.F., Kozhemyako, V.P. et al., "Method for training of a parallel-hierarchical network, based on population coding for processing of extended laser paths images," Proceedings of SPIE 4790, 465-478 (2002).

[33] Kirichenko, M.F., Krak, Yu.V., Polishchuk, A.A. "Pseudo inverse and projective matrices in problems of synthesis of functional transformers," Kibernetika i Sistemnyj Analiz 40 (3), pp. 116-129. ISSN: 00231274 (2004).

[34] Sawicki, D., "Using the GPU to determining the area the flame in the vision diagnostic system," IAPGOŚ 5(1), 80-85 (2015).

[35] Mazur, K., Wydra, M., Ksiezopolski, B., "Secure and Time-Aware Communication of Wireless Sensors Monitoring Overhead Transmission Lines," Sensors 17(7), 1640 (2017).

[36] Krak, Yu.V., "Dynamics of manipulation robots: Numerical-analytical method of formation and investigation of computational complexity," Journal of Automation and Information Sciences 31 (1-3), 121-128 (1999).