



DOI: 10.34910/MCE.109.10

Algorithm for building structures optimization based on Lagrangian functions

T.L. Dmitrieva^{a*} , Kh. Ulambayar^b 

^a Irkutsk National Research Technical University, Irkutsk, Russia

^b Mongolian University of Science and Technology, Mongolia

*E-mail: dmitrievat@list.ru

Keywords: modified Lagrangian multiplier method, building structures, optimum design, steel beam, truss structure, finite element model

Abstract. A review of modern algorithms and optimization programs is presented, based on which it is concluded that there is no application software in the field of optimal design of building structures. As part of solving this problem, the authors proposed numerical optimization algorithms based on conditionally extreme methods of mathematical programming. The problem of conditional minimization is reduced to a problem of an unconditional extreme using two modified Lagrange functions. The advantage of the proposed methodology is a wide range of convergence, the absence of requirements for convexity of functions on an admissible set of variation parameters, as well as high convergence, which can be achieved by adjusting the parameters of the objective and constraint functions. Verification of the developed methodology was carried out by solving a well-known example of ten-bar truss optimization. A comparison of the results obtained by other sources with the copyright ones confirmed the effectiveness of the presented algorithms. As an example, the problems of optimizing the cross-section of a steel beam have also been solved. Automation of the algorithms is performed in mathematical package *MathCAD*, which allows you to visually trace the sequence of commands, as well as obtain graphs that reflect the state of the task at each iteration. Thus, the authors obtained an original methodology for solving the optimization problem of flat bar structures, which can be extended to solve the problem of optimal design of general structures, where the optimality criterion is defined as material consumption, and the given structural requirements are presented as constraint functions.

1. Introduction

Most of these problems are formulated in the form of a nonlinear programming problem and were solved using gradient methods of the 1st or 2nd order or direct (zero order) methods of constrain and unconstrain minimization [1].

For the first time, the most general statement of the optimization problem was proposed by L. Schmitt [2], where he indicated the admissibility of the application of structural analysis using the finite element model and the nonlinear programming method in the presence of various forms of constraints. In 1979, a monograph was published by American scientists E. Haug (Edward J. Haug) and J. S. Arora (Jasbir S. Arora) [3]. This work gave a serious impetus to the development of the applied direction of optimization. It outlined general approaches to solving the problems of analysis and synthesis of mechanical systems. The 70-80s of the last century also accounted for numerous software implementations of optimization algorithms.

Note that the nonlinear programming methods embedded in these algorithms have a rigorous mathematical justification of the convergence conditions, but are rather laborious in solving large-



dimensional problems when it is necessary to optimize, for example, the topology of plane and spatial objects, explore dynamic processes, etc.

Nevertheless, these algorithms still find application in applied optimization problems, as well as in their software implementations [4, 5]. Wherein, the problem of static and dynamic analysis of structural is performed using the finite element method algorithm, which over the past decades has been developed for tasks of a special kind [6, 7].

Since the 90s, a large number of studies have appeared in the field of optimization of engineering systems that use metaheuristics. Metaheuristic algorithms explore the search space using probabilistic transition rules [8-10]. Here, a possible solution is sought by random selection, combination and variation of the desired parameters with the implementation of mechanisms resembling biological evolution, or physical processes occurring in nature. Here are just some examples of heuristic algorithms: genetic algorithm (GA) [11], ant colony algorithm (ACO) [12], artificial bee colony algorithm (ABC) [13], particle swarm optimization (PSO) [14], firefly algorithm (FA) [15], raven search algorithm (CSA) [16], gray wolf algorithm (AGW) [17], bat algorithm (BA) [18], annealing simulation algorithm (SA) [19] and others [20–22]. In [23–28], practical problems of optimizing building structures using metaheuristic algorithms are presented.

Most theoretical research in the field of numerical optimization of structures was accompanied by their software implementation. Among the leading software systems, including optimization modules, the “heavy” multipurpose universal software packages “MSC Nastran”, “ANSYS” and “ABAQUS” should be noted first of all. At the same time, in spite of many approaches, the problem of solving the problems of optimizing complex technical objects in an acceptable time and with a given degree of error still remains relevant. Over the past decade, a new direction has emerged related to the construction of optimization models based on the use of neural networks, where the search process is controlled by intelligent systems [29].

In the Russian design, the most fully automated calculations related to structural optimization are implemented in the field of aircraft manufacturing (just call the “IOSO NM” software package). A sufficient number of publications are devoted to the issues of optimizing building structures, which describe algorithms and software developments [30], however, these programs are of a research or narrow applied nature. In general, it should be noted that the optimization of construction projects is not yet widely used in real design.

This is due to the fact that the designer, on the one hand, does not have to possess the intricacies of optimization algorithms, and at the same time, such calculations require concomitant “manual” control, which is expressed, for example, in setting some parameters that significantly affect the convergence in tracking local extremes, etc. Thus, the problem of constructing universal optimization algorithms with a wide area of convergence, and at the same time not requiring serious “tuning” to the task, is quite demanded, significantly determining the implementation of these algorithms in design practice.

2. Methods

In this paper, we propose an algorithm for the numerical optimization of flat rod systems. In its most general form, this problem can be formalized as a conditionally-extremal nonlinear programming problem (NLP) [31–34].

$$\text{minimize} \quad f(x), \quad x \in E^{nx}, \quad (1)$$

$$\text{subject to:} \quad g_j(x) \leq 0, \quad j = 1, 2 \dots m, \quad (2)$$

$$x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2 \dots nx, \quad (3)$$

where $f(x)$ and $\{x\}$ are the objective function of variable parameters and the vector of these parameters on the interval $\{x^L\} - \{x^U\}$, and $g(x)$ is the constraint functions in the form of equalities and inequalities.

The problem of constrain minimization of the function of many variables (1-3) can be reduced to the problem of unconstrain minimization using the Lagrangian function F_L

$$F_L = f(x) + \{Y\}^T \{g\}, \quad (4)$$

where $\{Y\}$ is the Lagrange multiplier vector. In this case, the solution of problem (1-3) coincides with the saddle point of the Lagrangian function, for the determination of which the condition of its stationarity of this function with respect to x and y is used:

$$\frac{\partial F_L}{\partial x_i} = 0, \quad (5)$$

$$\frac{\partial F_L}{\partial y_j} = 0, \quad j = 1, 2, \dots, m.$$

These conditions were obtained for the convex optimization problem by Kuhn and Tucker, and can serve as a check of a reliable optimum.

Since the problem is solved in the space of two vectors $\{X\}$ and $\{Y\}$, the vector $\{X\}$ is usually defined as a vector of direct variables, and the elements of the vector $\{Y\}$ are called dual variables.

One of the significant drawbacks of the Lagrangian function is that it is applicable to a limited class of convex, separable programming problems. To construct methods with a wider region of convergence and applicable for finding a local extremum in non-convex problems, it is necessary to introduce additional terms into the structure of the function, which lead to an optimum shift in the search iterations, thus modifying the standard Lagrangian function. Various versions of the methods of a modified Lagrangian function are described in [35–39].

The purpose of this article is to illustrate the effectiveness of the methods of a modified Lagrangian functions and algorithms based on them in solving problems of optimizing geometric parameters of flat bar structures by the criterion of the minimum of their volume when fulfilling the standard requirements for strength and rigidity. The stated goal involves solving the following tasks:

- investigation of the properties of a modified Lagrangian functions;
- development of numerical algorithms based on these functions;
- solving problems using the proposed methods;
- comparison of the efficiency of algorithms for the rate of convergence and error in the results.

We introduce two modified Lagrangian function F_P and F_M

$$F_P = k_f F_L + 0.5 \{g\}^T [\delta] [k] \{g\} + 0.5 k_f \{Y\}^T ([\delta] - [I]) \{\Delta Z\}, \quad (6)$$

$$F_M = k_f F_L (1 - \tau) - 0.5 \tau \left\{ \frac{\partial F_L}{\partial x} \right\}^T \left\{ \frac{\partial F_L}{\partial x} \right\}. \quad (7)$$

Here, $[I]$ is identity matrix; $[k]$ is diagonal matrix of penalty coefficient; k_f is normalization factor introduced to increase computational stability; τ is convergence control constant; ΔZ_j is constraint shear amount to an allowable area; $[\delta]$ is diagonal matrix of Boolean variables whose elements are determined from the condition:

$$\text{if } g_j + \Delta Z_j > 0, \text{ then } \delta_{jj} = 1, \text{ otherwise } \delta_{jj} = 0.$$

Thus, the function F_P can be interpreted as the sum consisting of the Lagrangian function and the penalty for violation of restrictions, shifted by an amount ΔZ to an allowable area.

The function F_M is the sum of the Lagrangian function and the penalty for not fulfilling the stationarity conditions at the point $\{X^t\}$. If $\{X^*\}$, $\{Y^*\}$ are solution of the problem (1-3), then the ratio is true:

$$\frac{1}{k_f^t} F_M (X^*, Y) \leq F_L (X^*, Y^*) \leq \frac{1}{k_f^t} F_P (X, Y^*).$$

An equal sign in this expression is possible only with $X = X^*$; $Y = Y^*$, since at the same time penal additives turn into zero.

Consider iterative algorithms using a modified Lagrangian functions, which at each iteration in turn include two main procedures:

- determination of the vector of direct variables $\{X^{t+1}\}$;
- definition of a vector of dual variables $\{Y^{t+1}\}$.

The iterative process terminated by the condition of convergence:

$$\left| \{X^{t+1}\} - \{X^t\} \right| \leq \varepsilon_x \left| \{X^t\} \right|, \quad \bar{g}_j \leq \varepsilon_g, \quad j=1, \dots, m^*, \quad (8)$$

or if the specified limit number of iterations is exceeded it_lim .

In expression (8), $\{\bar{g}\}$ is a vector of potentially active constraints of dimension m^* ; ε_x , ε_g are predetermined calculation error; t is iteration number.

2.1. Direct method for solving the problem NLP

Consider an iterative algorithm for solving problem (1-3), which operates only with a function $F_P(x, y)$ and reduces to finding the saddle point of this function from the condition

$$\max_{y \in E^m} \min_{x \in E^{nx}} F_P(x, y).$$

The most stable version of this algorithm is when direct variables are determined from the condition of minimizing the function F_P

$$\begin{aligned} \{X\} &\in \text{Arg min } F_P(X^t, Y^t) \\ \{X^L\} &\leq \{X\} \leq \{X^U\}, \end{aligned} \quad (9)$$

and the dual conditions from the equality of stationarity with respect to X are the Lagrangian functions F_L (expression (4)) F_P and functions (expression (6)), which after the corresponding transformations has the form:

$$y_j^{t+1} = \max \left(y_j^t + \frac{k_{jj}^t}{k_f^t} g(x^{t+1}) \right).$$

This algorithm is investigated in [30]. In a comparative analysis with another algorithm, we designate it as "direct", since the search for an extremum at iteration t here is carried out only by direct variables $\{X\}$.

3. Results and Discussion

Based on the foregoing, the authors propose a numerical algorithm to solve problems (1-3) using two modified Lagrangian functions – function (6) and function (7).

3.1. Combined method for solving the problem NLP

At each iteration of the algorithm, to find the vector $\{X^{t+1}\}$, the function F_P in X (6) is also minimized, and dual variables are determined directly through straight lines by maximizing the function F_M .

$$\begin{aligned} \{Y^{t+1}\} &\in \text{Arg max } F_M(X^t, Y^t) \\ Y &\in E^m. \end{aligned} \quad (10)$$

Expression (10) is more convenient to obtain explicitly. For this, we substitute the expression of the Lagrangian function F_L in function F_M (7). To maximize the function F_M with respect to y , we perform differentiation with respect to this variable and equate the result to zero:

$$\begin{aligned} \left\{ \frac{\partial F_M}{\partial y} \right\} &= k_f (1 - \tau) [\delta] \{g(x)\} - \\ -0.5\tau &\left[\left(2[\delta] \left[\frac{\partial g(x)}{\partial x} \right]^T \left\{ \frac{\partial f(x)}{\partial x} \right\} + 2[\delta] \left[\frac{\partial g(x)}{\partial x} \right]^T \left\{ \frac{\partial g(x)}{\partial x} \right\} [\delta] \{Y\} \right) \right] = 0. \end{aligned}$$

Expand the brackets and simplify this expression:

$$k_f(1-\tau)[\delta]\{g(x)\} - \tau[\delta]\left[\frac{\partial g(x)}{\partial x}\right]^T \left\{\frac{\partial f(x)}{\partial x}\right\} - \tau[\delta]\left[\frac{\partial g(x)}{\partial x}\right]^T \left\{\frac{\partial g(x)}{\partial x}\right\}[\delta]\{Y\} = 0. \quad (11)$$

Since the function F_M is quadratic, expression (11) is reduced to a symmetric system of linear algebraic equations:

$$[W]\{\bar{Y}\} = \{P\}, \quad (12)$$

where $[W]$ is a matrix whose dimension is equal to the number of active constraints ($m^* \times m^*$), $\{\bar{Y}\}$ is vector of dual variables reduced to dimension m^* . Each element of this matrix is determined by the product of vectors:

$$W_{ij} = \left\{\frac{\partial \bar{g}_i}{\partial x}\right\}^T \left\{\frac{\partial \bar{g}_j}{\partial x}\right\}. \quad (13)$$

In expression (13) a dash sign indicates that derivatives are taken only by active restrictions. Matrix dimension $\left[\frac{\partial \bar{g}}{\partial x}\right] - nx \times m^*$.

The element i of the vector P is formed by the expression:

$$P_i = -k_f \left\{\frac{\partial \bar{g}_i}{\partial x}\right\}^T \left\{\frac{\partial f}{\partial x}\right\} + \frac{(1-\tau)}{\tau} g_i. \quad (14)$$

In (14) vectors $\left\{\frac{\partial \bar{g}_i}{\partial x}\right\}, \left\{\frac{\partial f}{\partial x}\right\}$ have dimension nx , where nx is the number of variable parameters that belong to an admissible region ($x_i^L \leq x_i \leq x_i^U$).

A distinctive feature of the combined algorithm is that when solving a conditionally extremal problem on its basis, there is no requirement for accuracy in the search for direct variables, i.e. their calculation at the iteration may have an error, and reflect only a certain movement towards the optimum.

A mixed form of these algorithms is possible when, at the first iteration, if a good initial approximation of the variables $\{X^0\}$ is known and there are no recommendations for the assignment of dual variables, a combined approach is used, and at subsequent iterations, the dual variables are recalculated using the direct method.

3.2. Algorithm for solving conditional-extreme problems for NLP using the combined method

The sequence of operations of the above algorithms differs only in the calculation of dual variables.

Since in the combined method this procedure is much more complicated, we present here a flowchart of the algorithm of the combined method (Fig. 1).

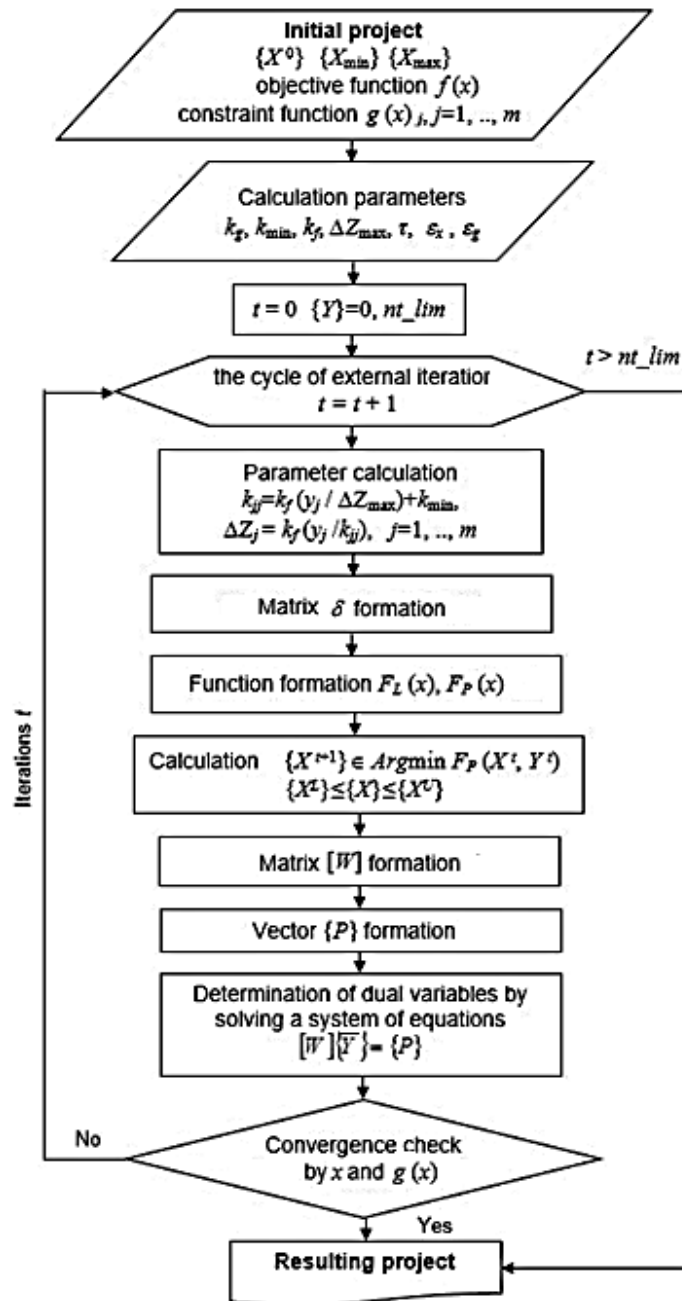


Figure 1. The flowchart of the algorithm for solving the conditional problem NLP.

3.3. Illustration of the proposed algorithm

Let us illustrate the operation of the algorithm by the example of the problem of optimal design of flat web steel I-beam, working on bending (Fig. 2).

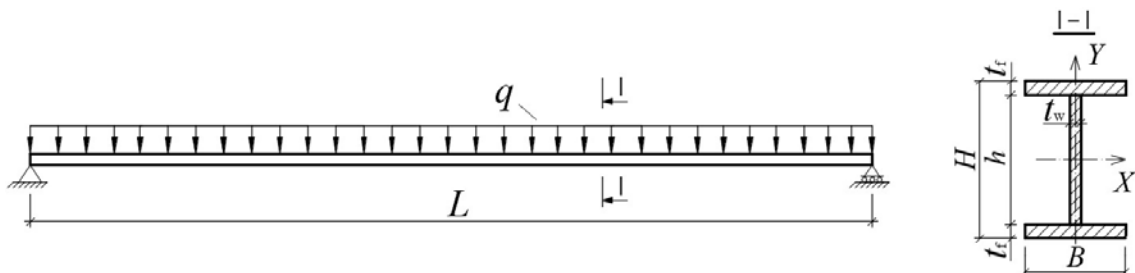


Figure 2. Calculation scheme of beam.

Initial data: $L = 9$ m; $q = 0.08$ MN/m.

Physical characteristics of the beam material: the modulus of elasticity $E = 2.06 \cdot 10^5$ MPa; the stress restrictions $R_u = 230$ MPa; $R_s = 133$ MPa.

Problem statement

It is required to select the parameters of the beam section at a given interval by minimizing its volume, provided that the regulatory requirements for strength and rigidity are met.

3 parameters of the cross section of the composite I-beam vary (Table 1).

Table 1. Variable parameters.

Parameter number	Designation	Description
X_1	h	Web depth
X_2	B	Flange width
X_3	$t_f; t_w$	Flange and Web thickness

The initial value of the varied parameters and the limits of their change:

$$X_1 = h = 0.50 \text{ m } [0.20 \div 1.20 \text{ m}]; X_2 = B = 0.1 \text{ m } [0.07 \div 0.60 \text{ m}]; X_3 = t_f = t_w = 0.03 \text{ m } [0.0008 \div 0.08 \text{ m}].$$

The objective function $f(x)$ represents the volume of the beam:

$$f(x) = (X_1 X_3 + 2X_2 X_3) L.$$

The restrictions used in the task are checks for strength and stability, and are set in accordance with the sections of Russian Construction Rules SP 16.13330.2001 "Steel structures". All restrictions reduced to dimensionless form are listed below.

The condition that ensures the local stability of the wall, in accordance with paragraph 8.5.1. the above Russian Construction Rules SP:

$$g_1 = \frac{h}{3.5t_f} \sqrt{\frac{R_u}{E}} - 1 \leq 0,$$

where R_u is the calculated resistance by yield strength.

The check on the strength by normal stress:

$$g_2 = \frac{M}{W_{n,\min} R_u} - 1 \leq 0,$$

where M and $W_{n,\min}$ are the calculated value of bending moment and the minimum moment of resistance.

The check on the strength by shear stress:

$$g_3 = \frac{QS}{Jt_f R_s} - 1 \leq 0,$$

where Q and R_s are the calculated value of shear force and the calculated resistance by shear, J is moment of inertia and S is the static moment of inertia.

The check by rigidity:

$$g_4 = \frac{\Delta_{\max}}{[\Delta]} - 1 \leq 0,$$

where Δ_{\max} and $[\Delta]$ are the maximum value of displacement in the beam and the maximum permissible value of displacement ($[\Delta] = L/500 = 0.018 \text{ m}$).

Thus, the number of variables is $n_x = 3$, and the number of constraints is $m = 4$.

The parameters of the optimization algorithm in the direct method were adopted as follows:

- minimum value of coefficient penalty (k_{\min}) - 80;

- maximum value of shear (ΔZ_{max}) - 0.2;
- normalization coefficient of the objective function (k_f) - $3 \cdot 10^6$;
- normalization coefficient of constraint functions (k_g) - 50.

Since the beam is statically determinable, the calculated forces do not change during the variation of the dimensions of the cross section.

For numerical solution, 2 iterative algorithms of modified Lagrangian function were applied – direct and combined.

Here is the sequence of operations of the direct method with the notation used in the MathCAD program:

1. Input of initial data:

1.1. Assignment of deterministic geometric (L) and physical (E , R_u , R_s) parameters of the problem, the value of the intensity of the uniformly distributed load q , as well as the limiting value of the displacement Δlim .

1.2. The purpose of the optimization algorithm parameters: the number of components of the constraint vector m ; normalization coefficients of the objective function and the constraint vector k_f , k_g ; the minimum value of the penalty coefficient k_{min} and the maximum value of the shift value outside the permissible range ΔZ_{max} .

1.3. Assignment of initial parameters of variation (vector X) and range of variation (vectors X_{min} and X_{max}).

1.4. The appointment of the initial values of the components of the vector of dual variables y_i ($i = 1 \dots m$).

2. Formalization of the objective function $f(X)$ and the vector of the constraint function $g(X)$ through variable parameters:

2.1. Formation of the objective function $f(X)$ through variable parameters.

2.2. Writing a subroutine for calculating the moment of inertia of the I-section $J(X)$.

2.3. Writing a subroutine for calculating the static moment of the I-section $S(X)$.

2.4. Formation of expressions of constraints g_1-g_4 by substituting in them the calculated values of the bending moment and shear force, which are expressed through variable parameters.

2.5. Formation of the constraint vector $g(X)$, where its components g_1-g_4 are multiplied by the normalization factor k_g .

3. Organization of a cycle where iterations of the search optimization algorithm are performed:

3.1. Assigning an it iteration counter.

3.2. Formation of a diagonal matrix of penalty coefficients by assigning its diagonal elements $k_{i,i}$ ($i = 1 \dots m$).

3.3. Formation of the elements of the shift vector of the admissible region of restrictions ΔZ_i ($I = 1..m$).

3.4. Formation of a diagonal matrix is a matrix of Boolean variables by assigning its elements $\delta_{i,i}$.

3.5. Formation of the Lagrange function $F_L(X)$, as well as the modified Lagrange function $F_p(X)$.

3.6. Formation of the block for finding the minimum of the function $F_p(X)$ by X on the interval X_{min} and X_{max} , where 2 search methods of unconstrained minimization are included: Conjugate Gradient method and Quasi-Newton method. When testing each of these, a practical coincidence of results was obtained.

3.7. Formation of the vector of values of the objective function at iterations F .

3.8. Identifying the maximum value in the vector of constraints on iterations it and entering it into the vector G .

3.9. Finding the ordinal number of the maximum element in the vector of constraints on iterations it and entering it into the vector k_k .

3.10. Calculation of the elements of the vector of dual variables y_i ($i = 1 \dots m$).

4. Output of results:

4.1. Numerical values of the vectors of change in the target and maximum values of the restrictive functions at iterations.

4.2. The values of the optimal parameters of the section of the beam X .

4.3. Values of residual constraints without taking into account the coefficient kg .

4.4. Graphs of changes in the target and maximum values of the restrictive functions at iterations.

4.5. Rounding of the beam section parameters to the required bit depth.

The solution was carried out in universal mathematical package *MathCAD*. *MathCAD* program listings are shown in Fig. 3–13.

The results of comparative calculations for the two methods are given in Table 2.

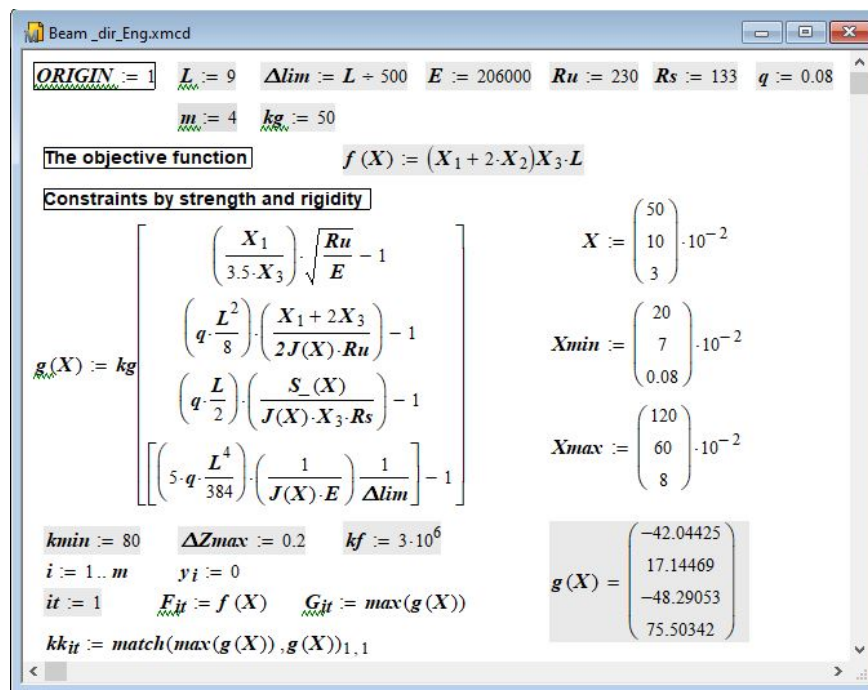


Figure 3. Implementing of numerical methods in mathematical package *MathCAD*: initial data of the task.

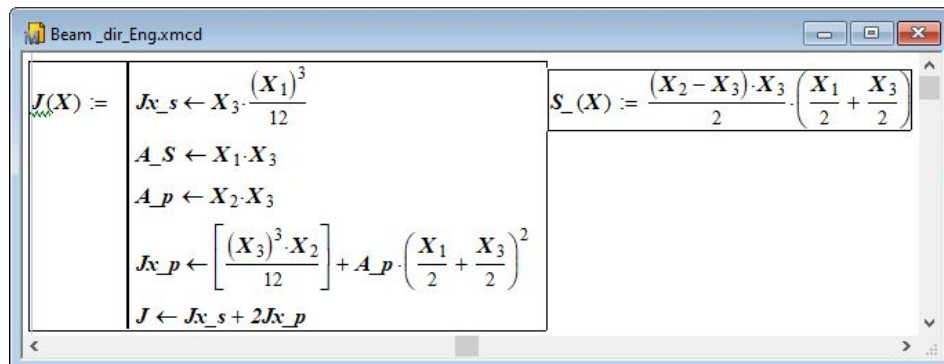


Figure 4. Subroutines used in the algorithm.

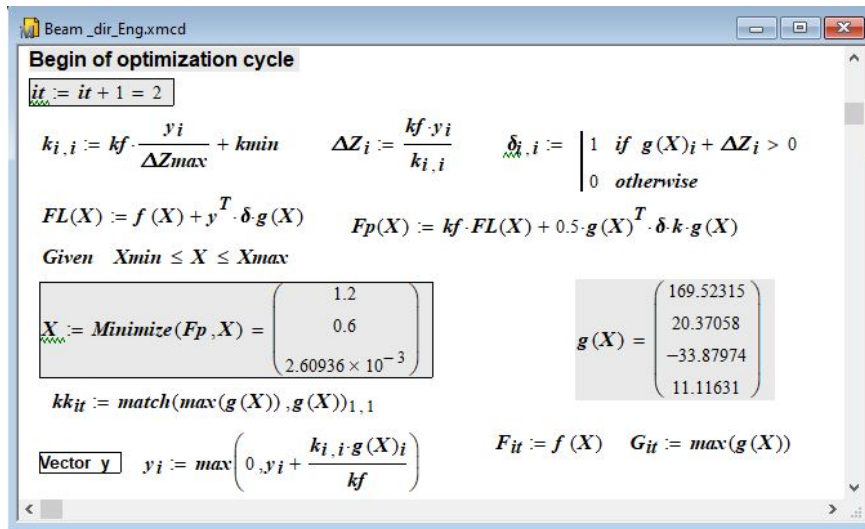


Figure 5. Begin of optimization cycle.

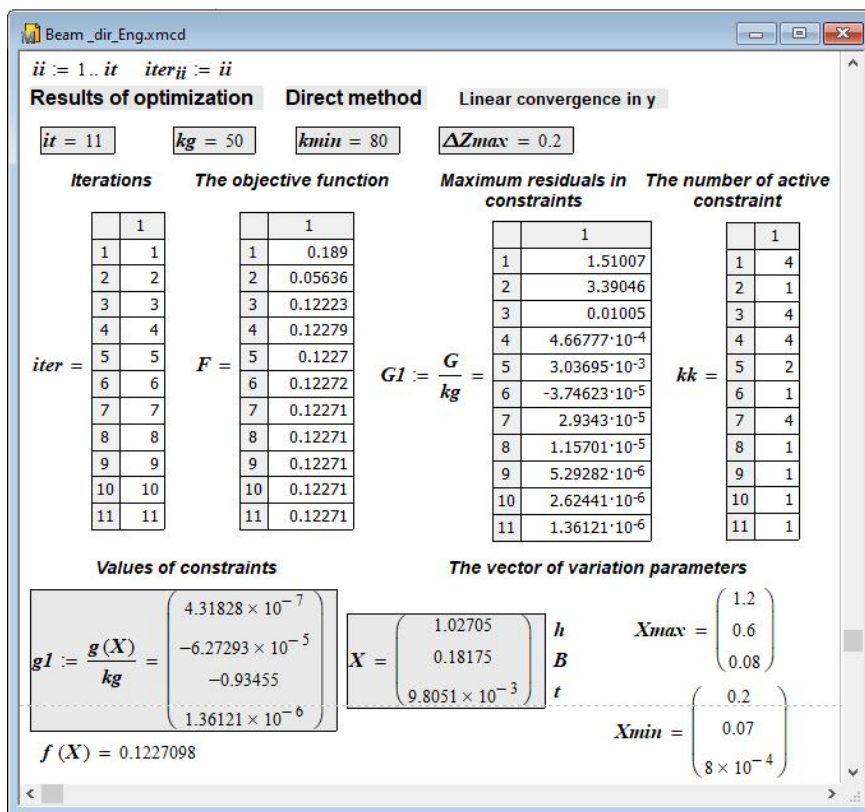


Figure 6. Optimization results.

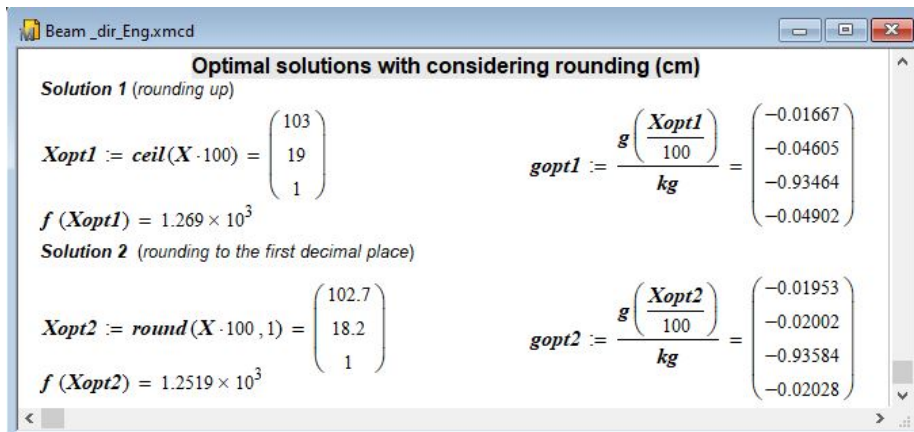


Figure 7. Optimal solutions with considering rounding.

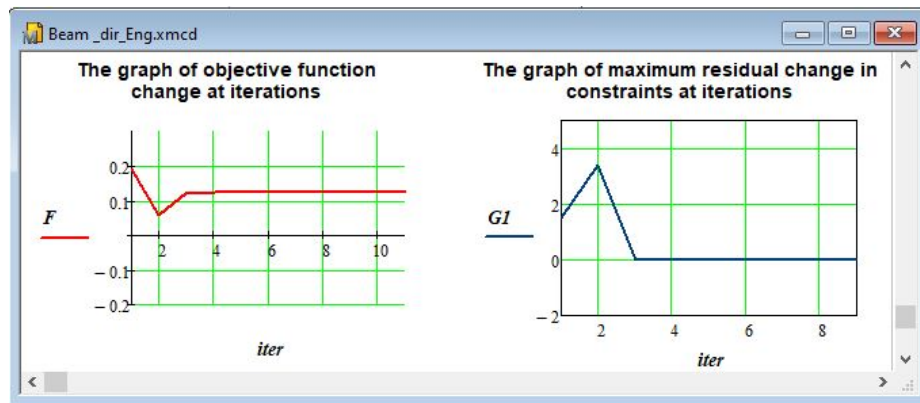


Figure 8. Graphs of changes in the objective function and maximum residuals of constraints at iterations.

The combined method algorithm differs in the following paragraphs:

P.1.2, where the parameters of the optimization algorithm are assigned, the assignment of the parameter τ is added, which is included in expression (14).

P.3.10, where the vector of dual variables $\{Y\}$ is formed, includes the solution of the system of equations (12). For this, the matrix $[W]$ and the vector $\{P\}$ are formed in the optimization cycle. This requires writing subroutines: dr_f (formation of the vector of derivatives of the objective function with respect to the varied parameters X) and dr_g (the formation of a matrix of derivatives of the functions of constraints with respect to the varied parameters X). In these subroutines, the input parameter Δ specifies the step of the offset from the current point in the numerical determination of derivatives.

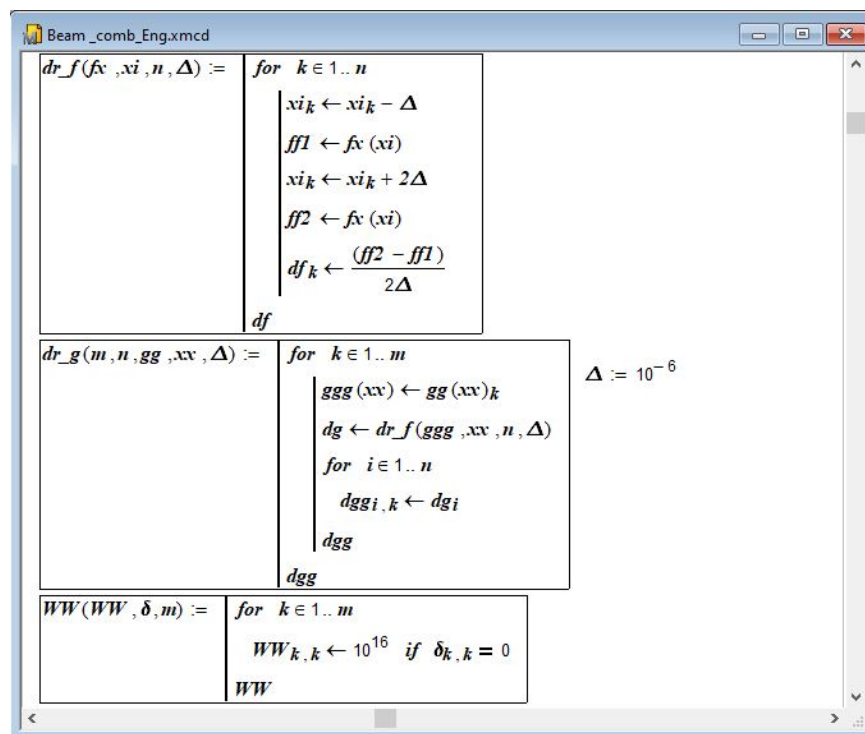


Figure 9. Subroutines used in the algorithm.

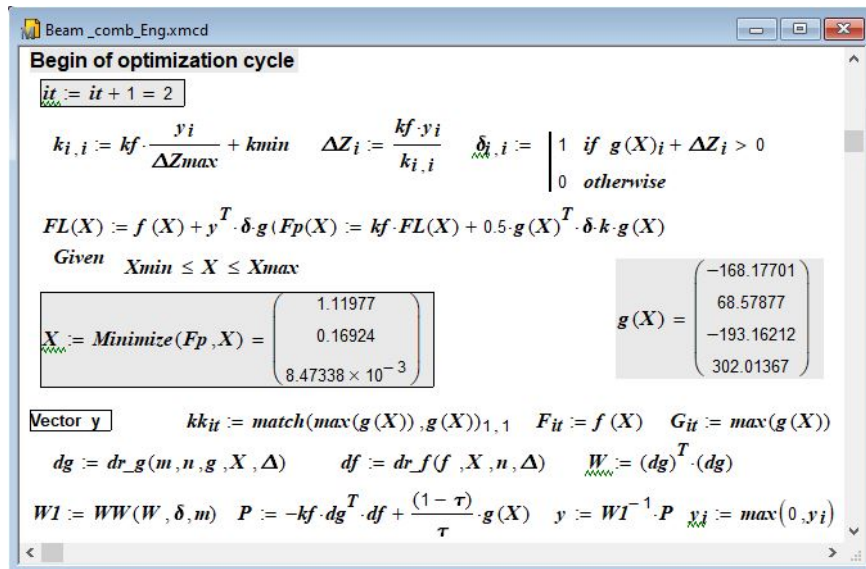


Figure 10. Begin of optimization cycle.

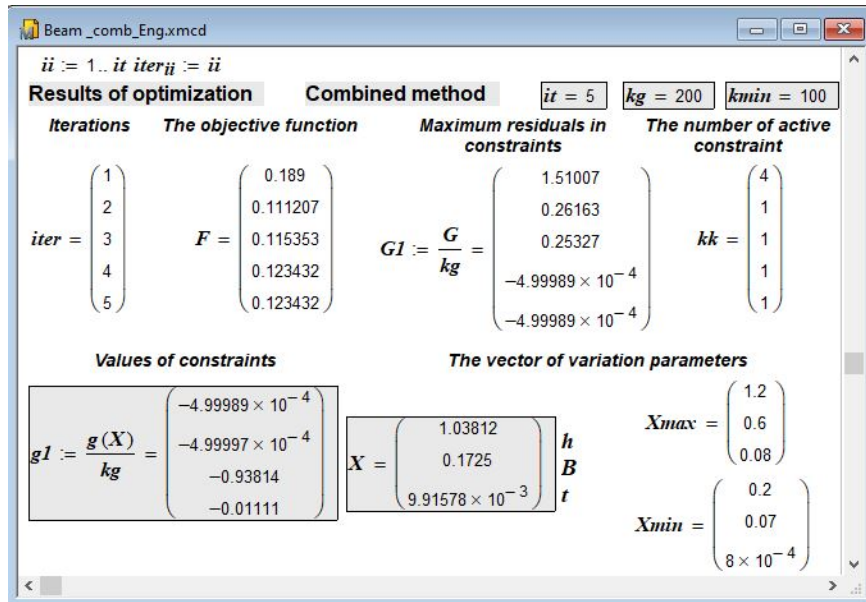


Figure 11. Optimization results.

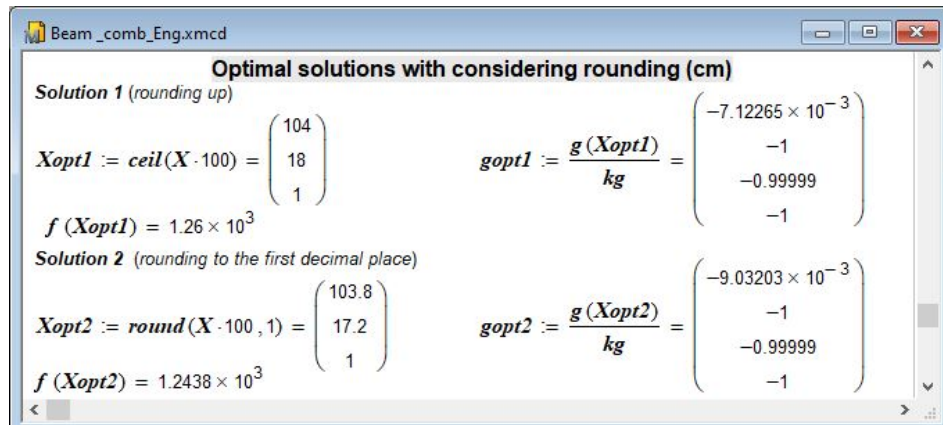


Figure 12. Optimal solutions with considering rounding.

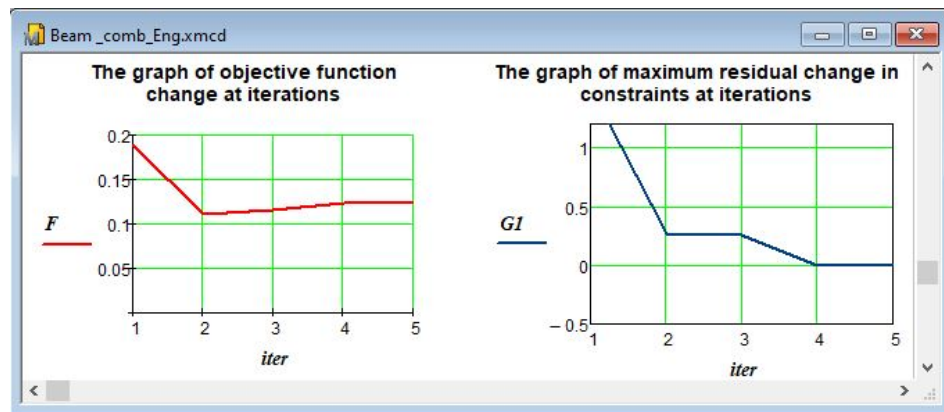


Figure 13. Graphs of changes in the objective function and maximum residuals of constraints at iterations.

Table 2. Comparative results of calculations.

№	Name	Numerical methods	
		Direct method	Combined method
1	Iteration number	11	5
2	Value of the objective function, m ³	$f(x) = 0.122791$	$f(x) = 0.123432$
3	Discrepancies in active constraints	$g_1 = 0.431828 \cdot 10^{-6}$ $g_2 = -0.627293 \cdot 10^{-4}$ $g_4 = 0.136121 \cdot 10^{-5}$	$g_1 = -0.499989 \cdot 10^{-3}$ $g_2 = -0.499989 \cdot 10^{-3}$
4	X_1 , m	1.02705	1.03812
5	X_2 , m	0.18175	0.1725
6	X_3 , m	$9.8051 \cdot 10^{-3}$	$9.91578 \cdot 10^{-3}$

The combined method showed better convergence with lower accuracy in constraint residuals, which does not change on subsequent iterations. It should be noted that the value of the objective function here is 0.6% higher than in the direct method.

The conditions for achieving the optimal solution at iterations, according to expression (8), were as follows:

- a) equality of variable parameters at 2 adjacent iterations;
- b) not exceeding the limits of the allowable area, taking into account the accepted errors for the constraint functions.

Thus, the accuracy of the solution was consistent with the accuracy in the calculation of potentially active constraints. The resulting solutions can be rounded to the required bit depth, which is reflected in the program listings.

4.1 Testing the proposed algorithm

The ten-bar truss optimization design problem is presented as verification of the proposed methodology. Let us consider a variant of this problem, set forth in the monograph [3] and article [40]. The geometry and loading conditions of ten-bar truss are shown in Fig. 14.

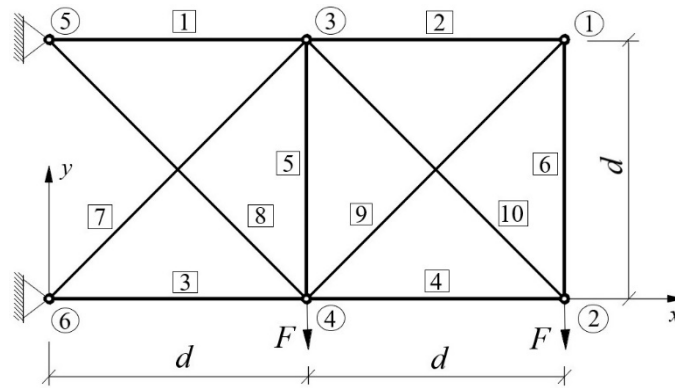


Figure 14. The geometry and applied loads for a ten-bar truss.

Initial data: $d = 9.144$ m (360 in); $F = 444.822$ kN (100 kips). All the members were constructed from a material with the modulus of elastic $E = 6.89476 \cdot 10^7$ kPa (10^4 ksi); the stress restrictions are $R_u = \pm 1.72369 \cdot 10^5$ kPa (± 25.0 ksi). The displacements for all nodes in all directions had to be less than $[\Delta] = \pm 0.0508$ m (± 2.00 in).

Problem statement. It is required to select the cross-sectional area of the truss at a given interval by minimizing its volume, provided that regulatory requirements for strength and stiffness are met.

The range of the available cross-sectional areas for each member in the truss vary between $A_{\min} = 6.4516 \cdot 10^{-5}$ m² (0.1 in²) and $A_{\max} = 0.02258$ m² (35 in²). Initial areas values $A_0 = 3.2258 \cdot 10^{-3}$ m² (5 in²).

The objective function $f(x)$ represents the weight of the truss.

The constraints are as follows:

The check on the strength in i element of the truss:

$$g_i = \frac{|N_i|}{R_u A_i} - 1 \leq 0, \quad i = 1, 2, \dots, 10.$$

In the test examples given in various sources, the calculation is performed without taking into account the buckling loss of compressed rods.

The constraint on vertical movement at joint 2:

$$g_{11} = \frac{\Delta_2}{[\Delta]} - 1 \leq 0.$$

Thus, the number of variables is $n_x = 10$ and the number of constraints is $m = 11$.

Were applied 2 iterative algorithms of modified Lagrangian function – direct and combined.

A feature of the implementation of the algorithms as applied to this problem was as follows:

1. Given that the truss is statically indeterminate, the analysis task was performed using the algorithm of the finite element method in displacements (FEA subroutine).
2. The functions of constraints on strength and stiffness contain state parameters (stress and displacement). Thus, the task of FE analysis, taking into account the changes in areas in the elements of the truss in this task, was recounted at each iteration.

Here is the sequence of operations of the direct method with the notation used in the MathCAD program:

1. Input of initial data:

11.1. Assignment of deterministic geometric (d) and physical (E , R_u) parameters of the problem, as well as the limiting value of displacements of nodes Δ_{lim} .

11.2. Assignment of truss geometry:

- setting the constants: ne (the number of members), np (the number of nodes), nop (the number of support reactions);
- setting vectors xp, yp (coordinates of nodes along the x and y axes);
- setting an array ni , which lists the connections of jointes for each members;
- specifying the vector iop , which lists the ordinal numbers of zero degrees of freedom connected by supports.

11.3. The purpose of the optimization algorithm parameters: the number of components of the constraint vector m ; normalization coefficients of the objective function and the constraint vector k_f, k_g ; the minimum value of the penalty coefficient k_{min} and the maximum value of the shift value outside the permissible range ΔZ_{max} .

1.8. Assignment of initial parameters of variation (vector X) and range of variation (vectors X_{min} and X_{max}).

1.9. The appointment of the initial values of the components of the vector of dual variables y_i ($i = 1 \dots m$).

2. Formalization of the objective function $f(X)$ and the vector of the constraint function $g(X)$ through variable parameters:

2.1. Writing the $FEA(X)$ subroutine that implements a finite element calculation of the truss. The output parameters of this subroutine are contained in an array, which includes three vectors – Z (nodal displacements), $Nabs$ (values of the internal forces in absolute value), L (lengths of elements).

Thus, the indices 1, 2, 3 that occur when calling this subroutine indicate which of these three arrays is being evaluated. Nodal displacements and internal forces are functions of variable parameters X (areas of sections of elements), therefore, they are defined as $Z(X)$ and $Nabs(X)$. The length of the elements L does not change in the course of optimization, therefore, it is calculated once.

2.6. Formation of the objective function $f(X)$ through variable parameters.

2.7. Formation of the constraint vector $g_1(X)$, which contains 10 strength checks for each truss element, multiplied by the normalization factor k_g .

2.8. Formation of the stiffness constraint $g_2(X)$, which contains a check for the vertical displacement of the truss node 2 (this displacement has index 4 in the general vector of nodal displacements Z). The constraint is also multiplied by the normalization factor k_g .

2.9. Formation of the complete constraint vector $g(X)$ by connecting the vectors $g_1(X)$ and $g_2(X)$.

This vector has 11 elements, where 10 corresponds to strength constraints and 11 corresponds to stiffness constraints.

3. Organization of a cycle where iterations of the search optimization algorithm are performed:

3.1. Assigning an it iteration counter.

3.2. Formation of a diagonal matrix of penalty coefficients by assigning its diagonal elements $k_{i,i}$ ($i = 1 \dots m$).

3.3. Formation of the elements of the shift vector of the admissible region of restrictions ΔZ_i ($i = 1 \dots m$).

3.4. Formation of a diagonal matrix is a matrix of Boolean variables by assigning its elements $\delta_{i,i}$.

3.5. Formation of the Lagrange function $F_L(X)$, as well as the modified Lagrange function $F_p(X)$.

3.6. Formation of the block for finding the minimum of the function $F_p(X)$ by X on the interval X_{min} and X_{max} ; The main difference in solving this problem is that each time the constraint function is accessed, the internal forces and displacements in the elements are recalculated, since function call $g(X)$ assumes calling in subroutine $FEA(X)$.

3.7. Formation of the vector of values of the objective function at iterations F .

3.8. Identifying the maximum value in the vector of constraints on iterations it and entering it into the vector G_i .

3.9. Finding the ordinal number of the maximum element in the vector of constraints on iterations it and entering it into the vector k_k .

3.10. Calculation of the elements of the vector of dual variables $y_i (i = 1...m)$.

4. Output of results:

4.1. Numerical values of the vectors of change in the target and maximum values of the restrictive functions at iterations.

4.6. Values of optimal areas of the truss elements (vector X).

4.7. Values of residual constraints without taking into account the coefficient kg .

4.8. Graphs of changes in the target and maximum values of the restrictive functions at iterations.

The results of the calculations are shown in Fig. 15-20. A comparison of solutions from sources [3, 40] with solutions obtained by authors is given in Table 3.

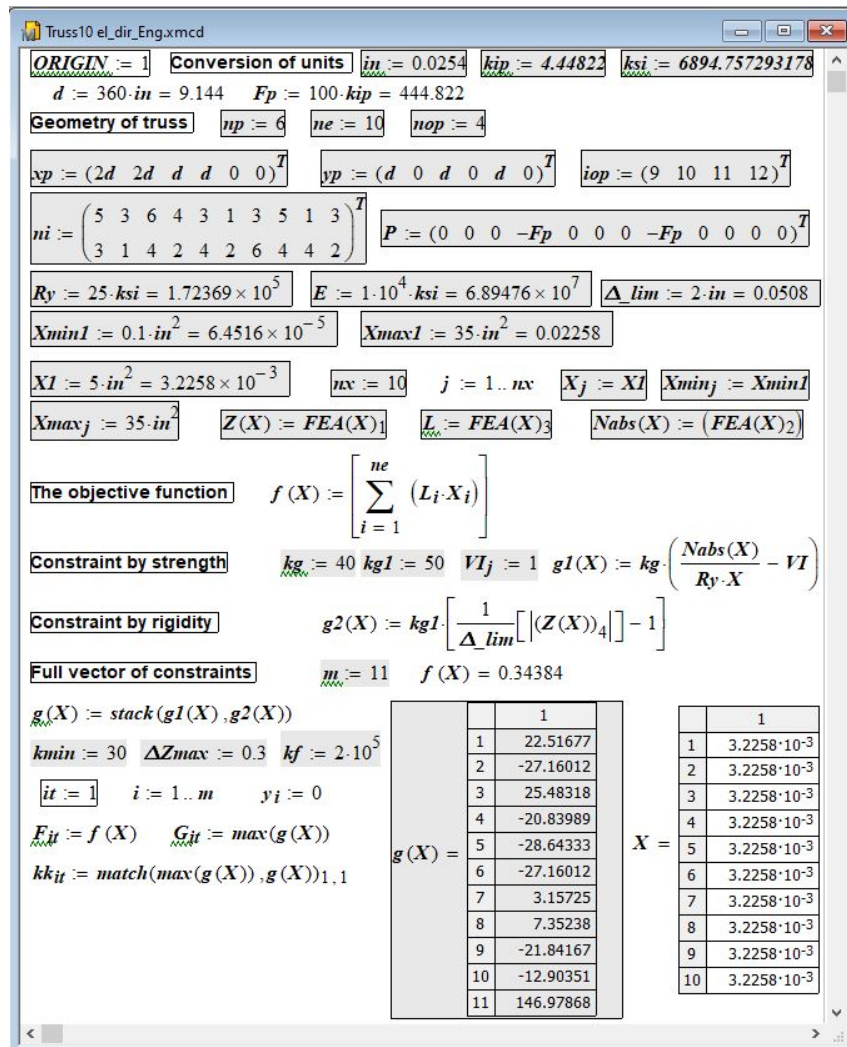


Figure 15. Implementing of numerical methods in mathematical package *MathCAD*: initial data of the task.


```

Truss10 e_dir_Eng.xmcd
Vabs(V,n) := for iv ∈ 1..n
              VIiv ← |VIiv|
              (VI)
FEA(X) := for i ∈ 1..ne
           lx_i ← xp(ni_i,2) - xp(ni_i,1)
           ly_i ← yp(ni_i,2) - yp(ni_i,1)
           Li ← √(lx_i^2 + ly_i^2)
           ci ← lx_i / Li
           si ← ly_i / Li
           i1 ← ni_i,1
           i2 ← ni_i,2
           A2_i1,i ← -si
           A2_i1-1,i ← -ci
           A2_i2,i ← si
           A2_i2-1,i ← ci
           Bi_i ← Li / E.X_i
           r ← A.B^-1.A^T
           for js ∈ (1..nop)
               r(iop_js,iop_js) ← 10^12
           Z ← r^-1.P
           N ← B^-1.A^T.Z
           Nabs ← Vabs(N,ne)_1
           ( Z
             Nabs
             L )
    
```

Figure 16. Subroutine FEA.

Truss10 e_dir_Eng.xmcd

ii := 1..it iterij := ii

Results of optimization Direct method

Iterations	The objective function	Maximum residuals in constraints	The number of active constraints
1	0.34384	3.67447	11
2	0.46038	1.05852	11
3	0.84033	0.01747	11
4	0.82457	-1.97597×10^{-3}	11
5	0.82311	-1.67874×10^{-5}	11
6	0.8231	6.52183×10^{-7}	5

$GI := \frac{G}{kg} =$

Values of constraints

	1
1	-0.73098
2	-0.95452
3	-0.65568
4	-0.73987
5	$-1.21465 \cdot 10^{-6}$
6	-0.95452
7	-0.72027
8	-0.25792
9	-0.93568
10	-0.73987
11	$6.52183 \cdot 10^{-7}$

The vector of variation parameters

	1
1	0.019435
2	$6.4516 \cdot 10^{-5}$
3	0.014795
4	$9.931763 \cdot 10^{-3}$
5	$6.425689 \cdot 10^{-5}$
6	$6.4516 \cdot 10^{-5}$
7	0.013387
8	$4.789997 \cdot 10^{-3}$
9	$6.4516 \cdot 10^{-5}$
10	0.014046

$X_{max} =$

	1
1	0.02258
2	0.02258
3	0.02258
4	0.02258
5	...

$X_{min} =$

	1
1	$6.4516 \cdot 10^{-5}$
2	$6.4516 \cdot 10^{-5}$
3	$6.4516 \cdot 10^{-5}$
4	$6.4516 \cdot 10^{-5}$
5	...

Figure 17. Optimization results.

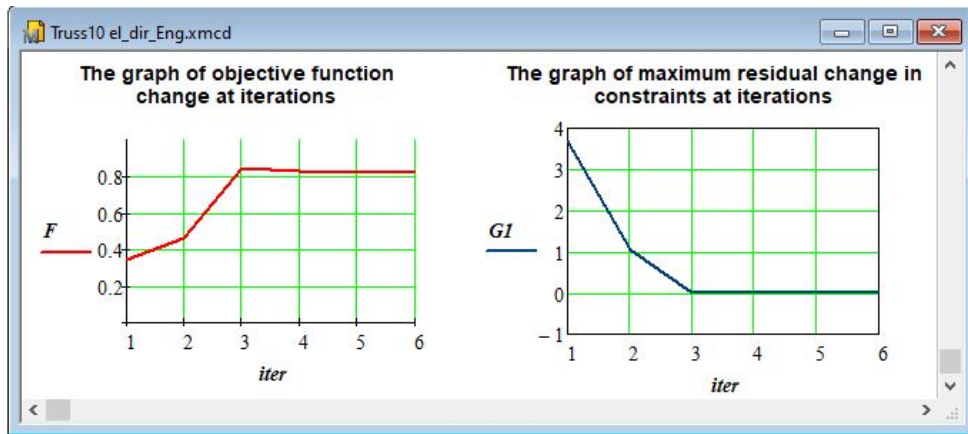


Figure 18. Graphs of changes in the objective function and maximum residuals of constraints at iterations.

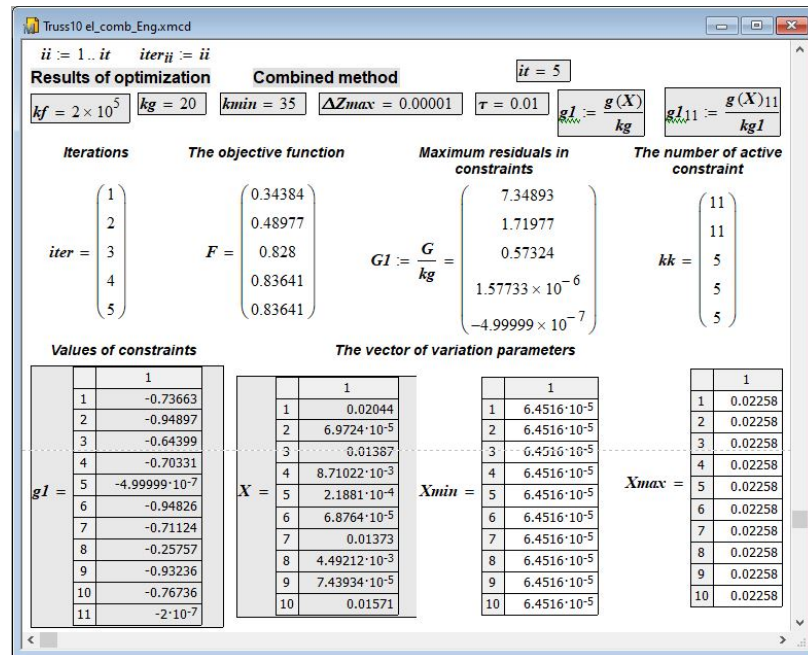


Figure 19. Optimization results.

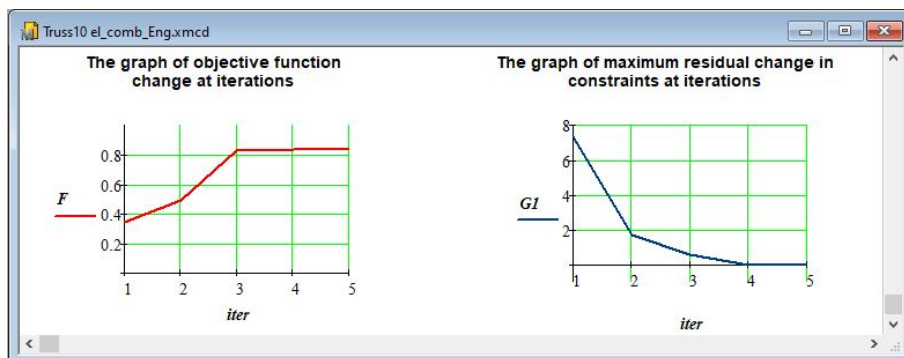


Figure 20. Graphs of changes in the objective function and maximum residuals of constraints at iterations.

Table 3. Comparative results of calculations.

№	Name	Numerical methods			
		Edward J. Haug [3]	B. Farchi [40]	Direct method	Combined method
1	Iteration number	15	–	6	5
2	Value of the objective function, $f(x)$, m ³	0.8294574	0.8294187	0.82312	0.83641
3	Discrepancies in active constraints, $g_5; g_{11}$	$-0.322 \times 10^{-4};$ -0.425×10^{-4}	$-0.9067 \times 10^{-4};$ -0.4315×10^{-4}	$-0.1215 \times 10^{-5};$ 0.6522×10^{-6}	$-0.4999 \times 10^{-6};$ -0.2×10^{-6}
4	$X_1 (A_1)$, m ²	$0.19374 \cdot 10^{-1}$	$0.19691 \cdot 10^{-1}$	$0.19435 \cdot 10^{-1}$	$0.2044 \cdot 10^{-1}$
5	$X_2 (A_2)$, m ²	$0.64516 \cdot 10^{-4}$	$0.64516 \cdot 10^{-4}$	$0.64516 \cdot 10^{-4}$	$0.69724 \cdot 10^{-4}$
6	$X_3 (A_3)$, m ²	$0.15015 \cdot 10^{-1}$	$0.14970 \cdot 10^{-1}$	$0.14795 \cdot 10^{-1}$	$0.13870 \cdot 10^{-1}$
7	$X_4 (A_4)$, m ²	$0.98619 \cdot 10^{-2}$	$0.98212 \cdot 10^{-2}$	$0.99318 \cdot 10^{-2}$	$0.87102 \cdot 10^{-2}$
8	$X_5 (A_5)$, m ²	$0.64516 \cdot 10^{-4}$	$0.64516 \cdot 10^{-4}$	$0.64516 \cdot 10^{-4}$	$0.21881 \cdot 10^{-3}$
9	$X_6 (A_6)$, m ²	$0.35903 \cdot 10^{-3}$	$0.35612 \cdot 10^{-3}$	$0.64516 \cdot 10^{-4}$	$0.68764 \cdot 10^{-4}$
10	$X_7 (A_7)$, m ²	$0.13676 \cdot 10^{-1}$	$0.13570 \cdot 10^{-1}$	$0.13387 \cdot 10^{-1}$	$0.13730 \cdot 10^{-1}$
11	$X_8 (A_8)$, m ²	$0.48182 \cdot 10^{-2}$	$0.48174 \cdot 10^{-2}$	$0.47899 \cdot 10^{-2}$	$0.44921 \cdot 10^{-2}$
12	$X_9 (A_9)$, m ²	$0.64516 \cdot 10^{-4}$	$0.64516 \cdot 10^{-4}$	$0.64516 \cdot 10^{-4}$	$0.74394 \cdot 10^{-4}$
13	$X_{10} (A_{10})$, m ²	$0.13947 \cdot 10^{-1}$	$0.13890 \cdot 10^{-1}$	$0.14046 \cdot 10^{-1}$	$0.15710 \cdot 10^{-1}$

The solution of the problem by the direct and combined method gave close results in terms of the number of iterations, the accuracy of the results and the nature of convergence, although in the combined method an overestimated value of the objective function was obtained by 0.8 %. Solutions close to optimal, were obtained already at the 3rd iteration with refinement at subsequent iterations to the required degree of accuracy. When comparing with the results obtained in two other sources, it should also be noted that the values in the areas are close with greater accuracy in the residuals of constraints (by 3-4 orders of magnitude).

In general, the results shown in the table show the high efficiency of the proposed algorithms, which give the best performance in almost all parameters compared with the results given in [3, 40].

The main difficulties in implementing the above examples were related to their adjustment to the task, which was carried out by setting optimization parameters, such as normalization coefficients of the target and restrictive functions, and the penalty coefficient. Guided by the experience of solving examples, we can relate the influence of these parameters on the convergence of the algorithm with the structure of the modified Lagrange function F_P . In the allowable region, this function coincides with the target, and outside the allowable values, the parameters k_g and k_{min} affect its curvature. It is revealed that the algorithm gives the most stable convergence if the values of the objective function are in the same order as the penalty additives.

4. Conclusions

In conclusion, we give the following conclusions:

1. The solution of practical examples of optimization of flat rod systems showed the effectiveness of the presented algorithms for searching for a constrain extremum. It was revealed that the solutions obtained using 2 algorithms of a modified Lagrangian functions in the first example gave a complete match, and in the second some discrepancy, which indicates the presence in the tasks of this class of several extremes.

2. Some difficulties associated with tuning algorithms to stable convergence are noted. If the objective function has a low order (volume in m³), and the constraints are normalized to unity, then it is necessary to increase the values of the objective function and penalty additions in the F_P function to a close order.

For this, the corresponding penalty and normalizing coefficients are introduced.

3. It is shown that the combined method for solving a conditionally extremal problem, although it has some complications in the definition of dual variables, can give better convergence. In addition, the

problems of searching for direct and dual variables in this algorithm are not interconnected. This is important when solving optimization problems of mechanics that have a complex formulation (for example, nonlinear problems, dynamics problems, etc.), when the problem of a conditional extremum in direct variables often becomes incompatible (especially in the initial iterations).

4. The study of the test problem of optimizing the ten-bar truss confirmed the efficiency of the developed algorithm. The results were obtained, giving the best indicators for the speed of convergence and accuracy compared with known solutions.

5. The optimization algorithm is presented in mathematical package *MathCAD*, which allows open access to its teams, confirms its adequacy and allows you to implement this algorithm to solve similar problems in design, scientific or educational activities.

6. In the software implementation of the practical problems of optimizing complex systems with a large number of variation parameters, it is more expedient to search for the unconditioned extremum using zero-order methods, such as the deformable polyhedron method, or based on metaheuristic algorithms.

7. Further research in the direction of optimization of planar and spatial rod systems can be effectively implemented by constructing optimization models based on neural networks.

References

- David, M.H. Applied nonlinear programming. McGraw-Hill. Texas, 1972. 498 p.
- Schmit, L.A. Structural Design by Systematic Synthesis. Proceedings, 2nd Conference on Electronic Computation, ASCE. 1960. Pp. 105–122.
- Edward, J. Haug., Jasbir, S.A. Applied optimal design: mechanical and structural systems. A Wiley-Interscience. Michigan, 1979. 506 p.
- Yamasaki, S., Kawamoto, A., Nomura, T., Fujita, K. A consistent grayscale-free topology optimization method using the level-set method and zero-level boundary tracking mesh. *International Journal for Numerical Methods in Engineering*. 2015. Vol. 101. Pp. 744–773. DOI: 10.1002/nme.4826
- Chen, J., Xiao, Z., Zheng, Y., Zheng, J., Li, C., Liang, K. Automatic sizing functions for unstructured surface mesh generation. *International Journal for Numerical Methods in Engineering*. 2017. Vol. 109. Pp. 577–608. DOI: 10.1002/nme.5298
- Lalin, V.V., Rybakov, V.A., Ivanov, S.S., Azarov, A.A. Mixed finite-element method in V.I. Slivker's semi-shear thin-walled bar theory. *Magazine of Civil Engineering*. 2019. 89 (5). Pp. 79–93. DOI: 10.18720/MCE.89.7
- Lalin, V.V., Yavarov, A.V., Orlova, E.S., Gulov, A.R. Application of the Finite Element Method for the Solution of Stability Problems of the Timoshenko Beam with Exact Shape Functions. *Power Technology and Engineering*. 2019. No. 53 (4). Pp. 449–454. DOI: 10.1007/s10749-019-01098-6
- Gandomi, A.H., Yang, X.S., Alavi, A.H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers*. 2013. No. 29 (1). Pp. 17–35. DOI: 10.1007/s00366-011-0241-y
- Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., Saka, M.P. Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Computers and Structures*. 2009. Vol. 87. Pp. 284–302. DOI: 10.1016/j.compstruc.2009.01.002
- Saka, M.P., Hasançebi, O., Geem, Z.W. Metaheuristics in structural optimization and discussions on harmony search algorithm. *Swarm and Evolutionary Computation*. 2016. Vol. 28. Pp. 88–97. DOI: 10.1016/j.swevo.2016.01.005
- Serpik, I.N., Alekseytsev, A.V. Optimization of flat steel frame and foundation posts system.
- Kaveh, A., Ghafari, M.H. Plastic analysis of planar frames using CBO and ECBO algorithms. *Iran University of Science & Technology*. 2015. No. 5 (4). Pp. 479–492.
- Aydın, D., Yavuz, G., Stützle, T. ABC-X: a generalized, automatically configurable artificial bee colony framework. *Swarm Intelligence*. 2017. No. 11 (1). Pp. 1–38. DOI: 10.1007/s11721-017-0131-z
- Bonyadi, M.R., Michalewicz, Z. Particle swarm optimization for single objective continuous space problems: A review. 2017. No. 25 (1). Pp. 1–54. DOI: 10.1162/evco_r_00180
- Nekouie, N., Yaghoobi, M. A new method in multimodal optimization based on firefly algorithm. *Artificial Intelligence Review*. 2016. No. 46 (2). Pp. 267–287. DOI: 10.1007/s10462-016-9463-0
- Askarzadeh, A.A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*. 2016. Vol. 169. Pp. 1–12. DOI: 10.1016/j.compstruc.2016.03.001
- Mirjalili, S., Mirjalili, S.M., Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software*. 2014. Vol. 69. Pp. 46–61. DOI: 10.1016/j.advengsoft.2013.12.007
- Cai, X., Wang, H., Cui, Z., Cai, J., Xue, Y., Wang, L. Bat algorithm with triangle-flipping strategy for numerical optimization. *International Journal of Machine Learning and Cybernetics*. 2017. No. 9 (2). Pp. 199–215. DOI: 10.1007/s13042-017-0739-8
- Samma, H., Mohamad-Saleh, J., Suandi, S.A., Lahasan, B. Q-learning-based simulated annealing algorithm for constrained engineering design problems. *Neural Computing and Applications*. 2019. Pp. 1–15. DOI: 10.1007/s00521-019-04008-z
- Cheng, M.-Y., Prayogo, D., Wu, Y.-W., Lukito, M.M. A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure. *Automation in Construction*. 2016. Vol. 69. Pp. 21–33. DOI: 10.1016/j.autcon.2016.05.023
- Jalili, S., Talatahari, S. Optimum Design of Truss Structures Under Frequency Constraints using Hybrid CSS-MBLS Algorithm. *KSCSE Journal of Civil Engineering*. 2017. No. 22 (5). Pp. 1840–1853. DOI: 10.1007/s12205-017-1407-y
- Rao, R.V., Saroj, A. Multi-objective design optimization of heat exchangers using elitist-Jaya algorithm. *Energy Systems*. 2016. No. 9 (2). Pp. 305–341. DOI: 10.1007/s12667-016-0221-9
- Lieu, Q.X., Do, D.T.T., Lee, J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Computers & Structures*. 2018. Vol. 195. Pp. 99–112. DOI: 10.1016/j.compstruc.2017.06.016

24. Degertekin, S.O., Hayalioglu, M.S. Optimum Design of Steel Space Frames: Tabu Search vs. Simulated Annealing and Genetic Algorithms. *International Journal Of Engineering & Applied Sciences*. 2009. No. 2 (1). Pp. 34–45.
25. Kaveh, A., Zakian, P. Improved GWO algorithm for optimal design of truss structures. *Engineering with Computers*. 2018. No. 34 (4). Pp. 685–707. DOI: 10.1007/s00366-017-0567-1
26. Jalili, S., Hosseinzadeh, Y. Combining Migration and Differential Evolution Strategies for Optimum Design of Truss Structures with Dynamic Constraints. *Iranian Journal of Science and Technology - Transactions of Civil Engineering*. 2019. No. 43 (1). Pp. 289–312. DOI: 10.1007/s40996-018-0165-5
27. Ho-Huu, V., Nguyen-Thoi, T., Vo-Duy, T., Nguyen-Trang, T. An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. *Computers and Structures*. 2016. Vol. 165. Pp. 59–75. DOI: 10.1016/j.compstruc.2015.11.014
28. Khatibinia, M., Yazdani, H. Accelerated multi-gravitational search algorithm for size optimization of truss structures. *Swarm and Evolutionary Computation*. 2018. Vol. 38. Pp. 109–119. DOI: 10.1016/j.swevo.2017.07.001
29. Donskoy, V.I. Extraction of Optimization Models from Data: an Application of Neural Networks. *Taurida Journal of Computer Science Theory and Mathematics*. 2018. No. 39 (2). Pp. 71–89.
30. Dmitrieva, T.L. Parametricheskaya optimizatsiya v proektirovanii konstruktssii, podverzhennykh staticheskomu i dinamicheskomu vozdeistviyu [Parametric optimization in the design of structures subject to static and dynamic effects]. *Izd-vo IrGTU. Irkutsk*, 2010. 176 p. (rus)
31. Fiacco, A.V., McCormick, G.P. Computational Algorithm for the Sequential Unconstrained Minimization Technique for Nonlinear Programming. *Management Science*. 1964. No. 10 (4). Pp. 601–617. DOI: 10.1287/mnsc.10.4.601
32. Allaire, G., Dapogny, C., Estevez, R., Faure, A., Michailidis, G. Structural optimization under overhang constraints imposed by additive manufacturing technologies. *Journal of Computational Physics*. 2017. Vol. 351. Pp. 295–328. DOI: 10.1016/j.jcp.2017.09.041
33. Chou, J.S., Ngo, N.T. Modified firefly algorithm for multidimensional optimization in structural design problems. *Structural and Multidisciplinary Optimization*. 2017. No. 55 (6). Pp. 2013–2028. DOI: 10.1007/s00158-016-1624-x
34. Golshtein, E.G., Tretyakov, N.V. *Modified Lagrangians and Monotone Maps in Optimization*. Wiley&Sons Publ. Co. New York, 1996. 438 p.
35. Bertsekas, D.P. *Constrained Optimization and Lagrange Multiplier Methods*. 1st ed. Athena Scientific. New York, 1996. 410 p.
36. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L. On augmented Lagrangian methods with general lower-level constraints. *SIAM Journal on Optimization*. 2007. No. 18 (4). Pp. 1286–1309. DOI: 10.1137/060654797
37. Wang, X., Zhang, H. An augmented Lagrangian affine scaling method for nonlinear programming. *Optimization Methods and Software*. 2015. No. 30 (5). Pp. 934–964. DOI: 10.1080/10556788.2015.1004332
38. Bertsekas, D.P. *Nonlinear Programming*. 3rd ed. Athena Scientific. New York, 2016. 861 p.
39. Long, W., Liang, X., Huang, Y., Chen, Y. A hybrid differential evolution augmented Lagrangian method for constrained numerical and engineering optimization. *CAD Computer Aided Design*. 2013. No. 45 (12). Pp. 1562–1574. DOI: 10.1016/j.cad.2013.07.007
40. Farshi, B., Alinia-Ziazi, A. Sizing optimization of truss structures by method of centers and force formulation. *International Journal of Solids and Structures*. 2010. No. 47 (18–19). Pp. 2508–2524. DOI: 10.1016/j.ijsolstr.2010.05.009

Contacts:

Tatiana Dmitrieva, dmitrievat@list.ru

Khukhuudei Ulambayar, Ulambayar_kh@yahoo.com