Informatics and Control Problems 43 Issue 1 (2023)
journal homepage: www.icp.az

On the application of modified dynamic programming in operational and organizational control

A.B. Sadigov

Institute of Control Systems, Baku, Azerbaijan

ARTICLE INFO

Article history:
Received 13.02.2023
Received in revised form 22.02.2023
Accepted 06.03.2023
Available online 05.04.2023

Keywords:
Operational and organizational control
Optimization
Modified dynamic programming

ABSTRACT

The methodology of application of dynamic programming and its modification in operational and organizational control is developed. To reduce dimensionality, computational difficulties, comparison of the methods and maximum use of all resources of computational means the principle of complexity is used. Computational procedures of dynamic programming are implemented in such a way that access to external computer memory is carried out as rarely as possible and at the same time all the output information is used effectively.

1. Introduction

The term "operational and organizational control" in this article means the processes of control and decision-making in organization systems and operational control systems based on the use of the prediction principle with periodic adjustments based on the processing of current in-formation, i.e., rolling plan principle.

The term "operational and organizational control" reflects the two most significant features of the investigated control processes.

The first feature is that the mathematical formalization of control problems of the control objects has constraints that are complex in form and various in content and do not fit entirely in the framework of known optimization problems, which requires taking into account a number of organizational measures.

The second feature has to do with the need for operational, i.e., prompt control decision making and reflects the fact of time shortage.

Problems of operational and organizational control are extremely widespread in the field of control, both for traditional dynamic objects and objects, the problems of optimal control of which have only begun to be studied in recent decades [1-8].

When implementing computational procedures of dynamic programming on a computer, the key factors limiting the applicability of the method are the amount of required memory and the computation time. Both of these factors stem from the dimensionality problem [9-15].

The question is whether it is possible to construct computational procedures for dynamic programming so that the external memory of the computer is accessed as rarely as possible, and at

E-mail address: aminaga.sadigov@gmail.com (A.B. Sadigov).

www.icp.az/2023/1-06.pdf https://doi.org/10.54381/icp.2023.1.06

the same time all the output information (not one number, but an array of numbers) is used effectively.

2. Prerequisites for the development of the method

This procedure can be constructed if the calculations are carried out in blocks and so that the operations are performed in each block independently. A possible way of dividing the grid of variables into blocks for the case when x is a scalar is shown in Fig. 1.

Suppose that the boundary condition at the left end is located on the boundary of the block $B\ (1,1)$, and at the right end — on the boundary of the block $B\ (4,4)$. Then the following order of calculating the blocks is convenient:

$$B\ (4,4) \to B\ (4,3) \to \cdots \to B(4,1) \to B(3,4) \to B\ (3,3) \to \cdots$$

$$\cdots \to B\ (3,1) \to B(2,4) \to B(2,3) \to \cdots \to B(1,4) \to \cdots \to B(1,1)$$

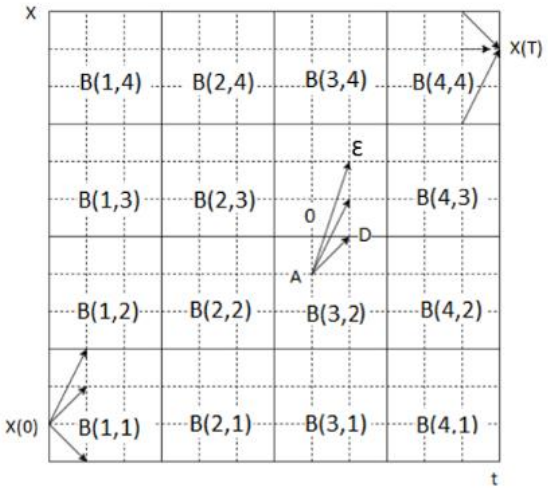

Fig. 1. Dividing the grid of variables into blocks

If the blocks are constructed rationally, then the information for calculating the trajectories inside each block can be derived from external memory devices for the entire block. This is possible only if the trajectories are located entirely inside the block and do not go outside its boundaries, as shown in Fig. 1 in block $B\ (3,2)$. However, in computational procedures of dynamic programming, for any method of dividing the grid of variables into blocks, there are almost always trajectories that go outside the block. If we limit ourselves to considering only the trajectories lying inside the block or ending at its boundaries, we risk losing the optimal solution. At the same time, we can try to organize the calculation process in such a way that the trajectories outside the block are analyzed at the points of their intersection with the boundary of the block.

It was proposed [16] to implement the block principle of the first computational procedure of dynamic programming by introducing a variable time quantization step, chosen so that the analyzed trajectory ends at one of the boundaries of the block, has a break at the intersection point, and continues along the block boundary to the nearest node of the grid of variables. (For instance, as shown in Fig. 1, the trajectory AOD should be used instead of the trajectory AC). In this case, the corresponding value of the Bellman function ω is determined on the basis of interpolation (or extrapolation) performed with respect to the values of $\omega_k(x_k)$ corresponding to the nodes of the grid of variables. This method has been called the "method of dynamic programming using increments of state variables" [16]. We could use a shorter name and call it the block dynamic programming method; however, following [17], we will call it the modified dynamic programming method.

3. Functional equations

We will consider the basic concepts of MDP using the example of already known Lagrange problem.

$$E\ (x,u) = \int_0^T F\ (x,u)dt\ \to min; \tag{1}$$

$$\dot{x} = f\ (x,u); \tag{2}$$

$$x\ (0) = a; \tag{3}$$

supplemented by the conditions

$$x\ (T) = b \text{ or not fixed}; \tag{4}$$

$$x\ (t) \in\ E_x; \tag{5}$$

$$u\ (t) \in\ E_u, \tag{6}$$

where, as before, x is a state variable; u is the control action; $x\ (0)$ and $x\ (T)$ are the boundary conditions; E_x and E_u are the regions of admissible states and controls.

For this problem, we obtained functional equations and considered computational procedures of dynamic programming.

Here we take into account that a grid of variables was constructed with quantization steps of the state variables δ_t and time variables Δ. When using the modified dynamic programming method (MDP), the same grid of variables [18] is constructed, but in this case the time quantization step, which we denote by δ_t, can be equal to Δ only in special cases. Therefore, we will use the functional equation

$$\omega(x,T) = \min_u[F(x,u)\mathrm{O}_t + \omega(x,T), T - \mathrm{O}_t].$$

Consider that this equation is valid for any time moment $t \in [0,T]$ and for any quantization step O_t. In MDP, $\delta_t < \Delta$ is selected, therefore the difference equation that is valid for Δ will also be valid for δ_t. It has the form

$$x(t+\delta_t) = x(t) + \delta_t f(x(t),u(t)). \tag{7}$$

Now we can write the functional equation in the following form:

$$\omega(x,T) - \min_u[F(x(t),u(t))\delta_t + \omega(x(t+\delta_t), t+\delta_t)]. \tag{8}$$

To implement MDP, as already mentioned, the same grid of variables is constructed as for the first computational procedure of dynamic programming, i.e., with quantization steps δ_t and Δ. In addition, the control quantization is carried out in a similar way, i.e., with step ν. Thus, constraints for problem (1)-(6) practically have the same form:

$$x(t) \in \{M^-, M^- + \delta, M^- + 2\delta, \dots, M^1 - 2\delta, M^1 - \delta, M^1\}; \tag{9}$$

$$u(t) \in \{M_1^-, M_1^- + \nu, M_1^- + 2\nu, \dots, M_1^+ - 2\nu, M_1^+ - \nu, M_1^+\}. \tag{10}$$

In MDP, the quantization step δ_t is not a constant but a variable, and $\delta_t \le \Delta$. In the cases when $\delta_t < \Delta$, it should be noted that $x(t+\delta_t) = x(t) + \delta = x(t+\mu\Delta)$, where μ is a positive integer. Most often $\mu = 1$. This is explained by the main condition of MDP: for any $u(t)$ from (10), the increment of the variable should be more than δ. Therefore, equation (8) can now be written in the form

$$\omega(x,T) - \min_{u}\left[F\left(x(t),u(t)\right)\delta_i \,/\, \nabla E + \omega(x(t+\mu\Delta),t+\mu\Delta)\right]. \quad (11)$$

Here the quantity ∇E is the increment of criterion (1) caused by the transition of the system from the state $x(t+\delta_t)$ to the state $x(t+\mu\Delta)$.. These states are equal in magnitude and differ only in the time instants $t+\delta_t$ and $t+\mu\Delta$. For instance, for the case shown in Fig. 1, the state $x(t)$ is determined by the point A, the state $x(t+\delta_t)$ — by the point Q, the state $x(t+\mu\Delta)$ — by the point D_1, i.e., $\mu = 1$. The ∇E value is determined by the trajectory OD. Since in MDP $x(t+\delta_t) = x(t+\mu\Delta)$, it usually turns out that $u(t)$ for $t \in [t+\delta_t, t+\mu\Delta]$ is zero. In these cases

$$\nabla E = F\left(x(t+\delta_t),u(t+\delta_t)\right)(\mu\Delta - \delta_t) \quad (12)$$

and $u(t+\delta_t) - 0$ on the entire interval $t \in [t+\delta_t, t+\mu\Delta]$.

However, as a rule, interpolation formulas are used in MDP to calculate ∇E, since the validity of (12) is not observed for some types of problems.

4. Selection of the time quantization step

The key concept that determines the essence of the MDP method is the concept of the block structure of the solution, which makes it possible to reduce the number of accesses to external storage devices of the computer, thereby significantly reducing the problem solution time. For this purpose, after dividing the grid of variables into blocks, it is necessary to provide special measures so that for each value $\vec{x}(t)$ the trajectory connecting $\vec{x}(t)$ and $\vec{x}(t+\Delta)$, does not go outside the boundaries of the block.

These measures in MDP consist in choosing the variable quantization step δ_t. The basic rule for selecting δ_t is that the value of δ_t should be such that the states $\vec{x}(t)$ and $\vec{x}(t+\Delta)$ differ from each other in each of their components by no more than a quantization step δ_t. This requirement implies the following formula for determining δ_t:

$$\delta_t = \min\left\{\frac{\delta_t}{|f_j(\vec{x},\vec{u},t_j)|}\right\}, \quad (13)$$

where δ_t is the quantization step of $x_j (j = \overline{1,n})$: f_j is the right-hand sides of the equations in system (2) written in the Cauchy normal form.

Formula (13) can be obtained for the scalar case from equation (7) if we solve it with respect to δ_t and assume in it that $x(t+\delta_t) - x(t) = \delta$. Then

$$\delta_t = \frac{\delta}{|f(x(t),u(t))|} \quad (14)$$

Since the values $f(x(t),u(t)$ can be both positive and negative, and $\delta_t > 0$, in the last equation and in (13) $f(\vec{x},\vec{u},t)$ is taken modulo. In Fig. 2, the selection of δ_t is explained graphically for $f(x(t),u(t) > 0$ —trajectory OA and for $f(x(t),u(t) < 0$ — trajectory OB.

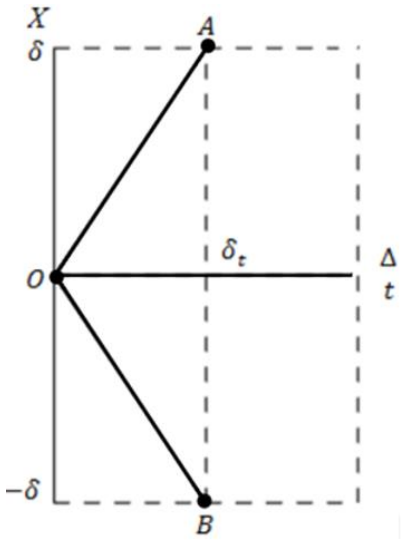

Fig. 2. Selecting the quantization step δ_t

5. Interpolation formulas

In Fig. 3 the corresponding values of $x(N\Delta)$ and $x\big((N-1)\Delta\big)$ are indicated by circles. Suppose that in the state $x=\delta$ at $t=(N-2)\Delta$, the control $u^{(1)}$ is used, which determines $\delta_t^{(1)}$ and the trajectory OA.

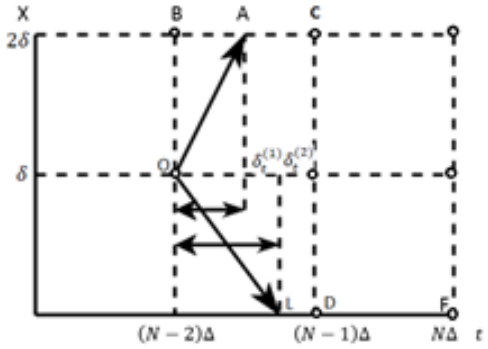

Fig. 3. Selecting interpolation formulas

Suppose also that the value of $\omega(2\delta,(N-2)\delta)$ has already been calculated. This node in Fig. 3 is denoted by the letter B. Then, to calculate ∇E corresponding to the trajectory AC, one can use the simplest interpolation formula connecting the value of ω for the nodes $\bar{B}$ and $\bar{C}$:

$$\nabla E=\frac{\omega(2\delta,(N-1)\Delta j-\omega(2\delta,(N-2)\Delta)}{\Delta}\delta_t^{(1)}.$$

In the general case, this formula for any values of x and t can be written as

$$\nabla E=\frac{\omega(x+\delta,t+\Delta)-\omega(x+\delta,t)}{\Delta}\delta_t, \qquad (15)$$

where the state x and the instant t correspond to the node of the grid of variables under investigation, and $(x+\delta)$ corresponds to the state at which the system will arrive from the state x after the time δ_t. Formula (15) is written for $f(x(t),u(t))>0$. Otherwise, $x+\delta$ in it should be replaced with $x-\delta$.

Taking into account interpolation (15), the value of $\omega(x,t)$ is determined from the formula

$$\omega(x,t) = \min_{u}[F(x(t),u(t))\delta_t + \nabla E + \omega(x+\delta,t)].$$

Suppose now that for the state $x+\delta$ corresponding to the instant t of application of the control u, the value of $\omega(x+\delta,t)$ has not yet been determined. Let, for instance, the value of $\omega(0,(N-2)\Delta)$ not be calculated yet for the trajectory OL shown in Fig. 3. Then interpolation can be carried out at the nodes corresponding to the points D and F:

$$\Delta E = \frac{\omega(0,(N-1)\Delta) - \omega(0,N\Delta)}{\Delta}\left(\Delta - \delta_t^{(2)}\right)$$

or in the general case

$$\Delta E = \frac{\omega(x \mid \delta_t \;\; t \mid \Delta) - \omega(x+\delta_t \;\; t \mid 2\,\Delta)}{\Delta}(\Delta - \delta_t). \qquad (16)$$

Taking into account (16), the value of $\omega(x,t)$ is determined as

$$\omega(x,t) = \min_{u}[F(x(t),u(t))\delta_t + \nabla E + \omega(x+\delta,t+\Delta)]$$

6. Computational procedure of modified dynamic programming

The MDP method uses the block principle of solution [19] in order to reduce the number of accesses to external storage devices of the computer and to carry out all calculations within each block using only random-access memory. The main requirement for the volume of each block follows from this: a block must "fit" into RAM entirely, together with the computation program and all the information necessary to carry out machine calculation. Usually, blocks with the number of nodes for each variable from 5 to 15 are formed.

After the blocks are formed, the important point is to determine the order in which the blocks are calculated. For a two-point boundary value problem, it is convenient to start computations from the block where the boundary conditions are located at the right end of the trajectory and end the computations with the block that contains the boundary conditions at the left end. Generally speaking, the blocks can be calculated almost in any order, but the labor intensity of the calculations decreases, and the accuracy of the results increases if such an order is chosen that the information of the previous block can be used as much as possible when calculating the current block. In this case, the number of extrapolations of the values of $\omega(x+\delta,t+\Delta)$ for the $x+\delta$ and $t+\Delta$ that are located in previously uncalculated blocks decreases dramatically.

There is one more possibility of reducing the number of extrapolations of the values of $\omega(x+\delta,t+\Delta)$.. Suppose that, as shown in Fig. 4, the state $x(t)$ belongs to the boundary of blocks $B(p,l)$ and $D(p,l+1)$, of which block $D(p,l)$ is being calculated at the moment, and block $B(p,l+1)$ had not yet been calculated.

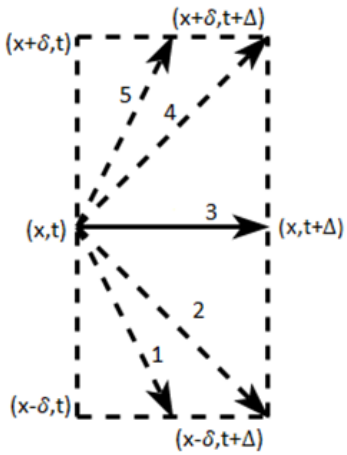

Fig. 4. Illustration of the method of correction of $\omega(x,t)$

Suppose that the set of admissible controls determines five trajectories, of which trajectories 1, 2, and 3 can be estimated from the information available in block $B(p,l)$, and to estimate trajectories 4 and 5, it is necessary to extrapolate the values of $\omega(x+\delta,t)$ and $\omega(x+\delta,t+\Delta)$. Choose the optimal trajectory from trajectories 1, 2, and 3 and fix the corresponding value of $\omega'(x,t)$ in the block $B(p,l)$.. Further, when calculating the block $B(p,l+1)$, we will correct $\omega'(x,t)$, by choosing the best one from trajectories 4 and 5 and comparing the obtained value of $\omega''(x,t)$ with $\omega'(x,t)$. Unfortunately, if $\omega'' < \omega'$, this method requires correction not only of the values of $\omega(x,t)$, but also of all subsequent values of ω calculated using the uncorrected value of $\omega(x,t)$.

7. An example of using the modified dynamic programming method

As an example of using MDP, consider the following problem:

$$\left.\begin{gathered} E\,(x,u) = \int_0^T (x^2 + u^2)dt + \lambda\,[x\,(T) - \beta]^2 \to min; \\ \dot{x} = u; \\ u \in E_u = [u: -2,-1,0,1,2]; \\ x_{min} \le x\,(t) \le x_{max}; \\ 0 \le x(t) \le \beta. \end{gathered}\right\} \quad (17)$$

Suppose, as before, that $T = 10$, $\lambda = 2.5$, $\beta = 2, x_{min} = M^- = 0$, $x_{max} = M^+ = 8, \Delta = 1, \delta = 1$. Then problem (17) can be reduced to the following mathematical programming problem:

$$\left.\begin{gathered} E(x_k,u_k) = \sum_{k=0}^{N-1-5} (r_k^2 + u_k^2) + 2{,}5\,[x_{10} - 2]^2 \to min; \\ x_{k+1} = x_k + u_k; \\ u_k \in E_u = (-2,-1,0,1,2); \\ 8 < x_2 < 8; \\ 0 < x_{11} < 2. \end{gathered}\right\}$$

Using the MDP method, we write this problem in the following form:

$$\left.\begin{array}{c} E(x,u)=\sum[x^2(t)+u^2(t)]+2{,}5\,[x(T)-2]^2\to min; \\ x(t+0_i)=x(t)+u(t)\delta_t; \\ u(t)\in[2,\quad 1,0,1,2]; \\ 2<x(t)<8; \\ 0<x(T)<2. \end{array}\right\} \tag{18}$$

where the time instants t correspond to the nodes of the grid of variables with a constant step Δ.

The step δ_t depends on the value of the control action applied at each time instant and is determined from general formula (13):

$$\delta_t = min\left\{\frac{\delta}{[f(x(t),u(t))]},\Delta\right\} = min\left\{\frac{\delta}{[u]},\Delta\right\}.$$

For $u=\pm 2$, we find $\delta_t = 0.5$; for the other values of u we obtain $\delta_t = \Delta = 1$.

When calculating $\delta_t = 0.5$, it is necessary to have the interpolation formulas for calculating ∇E. For this purpose, we will use formula (15) if the interpolated values of $\omega(x,t)$ and $\omega(u \mid \delta, t \mid \Delta)$ are already known, or formula (16), if the values of $\omega(x+\delta, t+\Delta)$ and $\omega(x+\delta, t+2\Delta)$ are known. Considering that $\delta = \Delta = 1$, formulas (15) and (16) for $u>0$ will take the form

$$\nabla E - (\omega(x+1,t+1) \quad \omega(x+1,t)\delta_t; \tag{19}$$

$$\nabla E - (\omega(x+1,t+1) - \ \omega(x+1,t+2)(1-\delta_t). \tag{20}$$

If $u<0$, $x+1$ in (19) and (20) should be replaced with $x-1$.

To calculate the values of $\omega(x+0, t+\mu\Delta) = \omega(x+1, t+\mu)$ for $\mu=1$ and $\mu=2$, when the state $x+1$ is in a neighboring block, for which calculations have not yet been performed, we will use extrapolation formula for $\mu=1$ and $\mu=2$ as follows:

$$\omega(x+1,t+1) = 3\omega(x,t+1) - 3\omega(x-1,t+1) + \omega(x-2,t+1); \tag{21}$$

$$\omega(x+1,t+2) = 3\omega(x,t+2) - 3\omega(x-1,t+2) + \omega(x-2,t+2); \tag{22}$$

Functional equation (11) is applicable to the optimization problem under consideration. Using the notation of problem (18), we obtain

$$\omega(x,t) = \min_{u(t)}\left[\left(x^2(t)+u^2(t)\right)\delta_t + \nabla E + \omega(x(t)+u(t)\delta_t, t+1)\right].$$

In this equation, x and t take on the discrete values $0, 1, 2, \dots, 8$ and $0, 1, 2, \dots, 10$, respectively, and

$$x(t)+u(t)\delta_t = \begin{cases} x(t)+1 & \text{when} \quad u>0; \\ x(t) & \text{when} \quad u=0; \\ x(t)=1 & \text{when} \quad u<0. \end{cases}$$

To solve the problem by the MDP method, we divide the grid of variables into eight blocks and take the following order of calculating the blocks:

$$B(2,1), B(2,2), B(2,3), B(2,4), B(1,1), \dots, B(1,4).$$

To store the optimal values of $\omega(x,t)$ and optimal controls $u(t)$, since the dimensionality of the problem is small, we form a single table for all blocks.

The calculation of the first block is preceded by a preliminary procedure that limits the search area by excluding invalid nodes from the grid of variables.

As a result of this procedure, we determine the values of $\omega(x,t)$ and the optimal controls for the boundary admissible nodes on the right side of the grid of variables. The values of $\omega(x,10)$ are

known for $0 \leq x \leq 2$; states $x \geq 3$ at $t = 10$ are invalid. For $t = 9$, the minimum state from which $x(10) = 2$ is attainable is $x(9) = 4$. In this case, $u = minE_u = -2$. The states $x(9) > 4$, are therefore invalid. For $t = 8$, the states $x(8) > 6$ are invalid. Let us calculate the values of $\omega(x,t)$ corresponding to this trajectory. Since the smallest value of δ_t is 0.5, we will carry out the calculations at $\delta_t = 0.5$ instead of $\delta_t = \Delta = 1$. Decreasing the quantization step will increase the accuracy of calculating $\omega(x,t)$.

In this case, there is no need to use the first computational procedure of dynamic programming for calculating $\omega(x,4)$, as we did for the bloc $B(2,1)$ when calculating $\omega(x,9)$.

8. Conclusion

Modified dynamic programming does not lead either to a reduction in the number of nodes in the grid of variables, or to a decrease in the computational difficulties of analyzing local trajectories. Moreover, the amount of computation in comparison with the first computational procedure of dynamic programming increases significantly, which is due to the need for interpolation and extrapolation. The number of nodes of the grid of variables in MDP, if summed over all blocks, remains exactly the same as in the first or second computational procedure, regardless of the method of dividing the problem into blocks. The effect of reducing the dimensionality will be observed only if the direction of the preferred movement is well known and there is no need to carry out calculations in many blocks.

Nevertheless, as shown in (18), the practical use of MDP leads to a significant reduction in the computing time in comparison with the first and second computational procedures, and this is explained by only one factor: the reduction in the number of accesses to external storage devices of the computer.

References

[1] А.Б. Садыгов, Модели и технологии решения задач управления в чрезвычайных ситуациях, Баку, «Элм», 2017, 372 p. [In Russian: A.B. Sadigov, Models and technologies for solving control problems in emergency situations, Baku, Elm]. https://www.researchgate.net/publication/315893563

[2] A.B. Sadigov, I.I. Mustafayev, G.N. Hajimatov, E.K. Gafarov, Description of a region as an object of control in emergency situations, Informatics and control problems. 41 No.2 (2021) pp. 3-7. https://www.doi.org/10.54381/icp.2021.2.01

[3] A.B. Sadigov, B.H.Salayev, S.S.Gezalov, Modeling human actions and psychophysical reactions in emergency situations, Informatics and control problems. 42 No.1 (2022) pp.3-10. https://doi.org/10.54381/icp.2022.1.01

[4] A.B. Sadigov, R.M. Zeynalov, Mathematical modeling of environmental processes at military facilities, Informatics and control problems. 40 No.2 (2020) p.31-37. https://www.icp.az/2020/2-04.php

[5] A.E. Bryson, Applied optimal control: Optimization, estimation and control, New York, (2017) 496 p. https://doi.org/10.1201/9781315137667

[6] V.V. Zakharov, Models and algorithms for planning the operation and modernization of a corporate information system based on a service-oriented approach, Dissertation for the degree of candidate of technical sciences, Saint Petersburg, (2021) 172 p. http://www.spiiras.nw.ru/dissovet/wp-content/uploads/2021/03/zahar ovvv_dissertacija.-18.03.2021-_novye-akty.pdf

[7] Sanford M Roberts, Dynamic programming in chemical engineering and process control, Elsevier, (1964) 322 p. https://books.google.az/books/about/Dynamic_Programming_in_Chemical_Engineer.html?id=3Y-4m3lVdgMC&redir_esc=y

[8] S.A. Orshansky, A.A. Shalyto, Application of dynamic programming in solving problems on finite automata, Computer tools in education. No.4 (2006) pp.26-35. [In Russian:]. https://is.ifmo.ru/works/_2007_09_10_orshanskiy.pdf

[9] A.B. Sadigov, R.M. Zeynalov, Optimal control in the problems of calculating the benefit/cost ratio in emergency response, Informatics and control problems. 40 No.1 (2020) p.47-56. https://icp.az/2020/1-06.pdf

[10] A.B. Sadigov, Methodological restrictions of the theory of ensuring safety of the ecologically dangerous objects, Intern. journal of research and reviews in applied sciences. 13 No.1 (2012) pp. 175-179.

https://www.arpapress.com/Volumes/Vol13Issue1/IJRRAS_13_1_19.pdf
[11] A.B. Sadigov, B.A. Guliyev, E.N. Sabziyev, The problem of optimal location of the military logistics center, Informatics and control problems. 41 No.1 (2021) pp.3-9. https://www.icp.az/2021/1-01.pdf
[12] Aminaga Sadigov, Turan Isazadeh, Lamiya Najafova, Dynamic models in operational and organizational control systems: Dynamic programming method, Proceedings of the international conference on problems of logistics, Management and operation in the east-west transport corridor (PLMO), October 27-29, 2021, pp. 232-241. https://conf.cyber.az/files/papers/PLMO_All_Papers.pdf
[13] Aminaga Sadigov, Ramin Zeynalov, Ramin Gasanov, Leyla Gadimova, Formulation of the problem and general requirements for optimization methods of operational and organizational control systems, Proceedings of the international conference on problems of logistics, Management and operation in the east-west transport corridor (PLMO), October 27-29, 2021, pp. 217-225. https://conf.cyber.az/files/papers/PLMO_All_Papers.pdf
[14] А.Б. Садыгов, Задача о назначениях в системах оперативно-организационного управления, The Reports of National Academy of Sciences of Azerbaijan. LXXVII No.3-4 (2022) pp.3-8. [In Russian: A.B. Sadigov, The problem of assignments in operational and organizational control systems] https://www.researchgate.net/publication/367523881_Zadaca_o_naznaceniah_v_sistemah_operativno-organizacionnogo_upravlenia
[15] I.I. Mustafayev, A.B. Sadigov, Modeling of regimes with sudden change, Reliability: theory and applications, USA. 17 No.4(70) (2022) pp.111-117. https://doi.org/10.24412/1932-2321-2022-470-111-117 [12]
[16] A.M. Korneev, T.V. Lavrukhina, T.A. Smetannikov, Solution of the problem of optimal distribution of resources of the Ministry of emergency situations by the method of dynamic programming, Trends in the development of science and education. 71 No.1 (2021) pp.20-24. [In Russian:]. https://elibrary.ru/item.asp?id=45538520
[17] Wanshan Wang, Peihong Wang, Yihua Dong, Modified dynamic programming algorithm and its application in distribution of power plant load, E3S Web of Conferences 136, 01005 (2019). https://www.e3s-conferences.org/articles/e3sconf/pdf/2019/62/e3sconf _icbte2019_01005.pdf
[18] L.G. Gauss, A. Zhao, Dynamic programming and applications, (2020) 16 p. https://math.mit.edu/research/highschool/primes/circle/documents/2020/GonzalezGauss_Zhao_2020.pdf
[19] V.I. Gurman, I.V. Rasina, I.S. Guseva, O.V. Fesko, Methods for approximate solution of optimal control problems, Program systems: theory and applications. No.4(27) (2015) pp.113-137. [In Russian:]. https://psta.psiras.ru/read/psta2015_4_113-137.pdf