DOI: http://doi.org/10.25728/cs.2022.6.1

DESIGN OF INTEGRATED RATING MECHANISMS BASED ON SEPARATING DECOMPOSITION

V.A. Sergeev

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ sergeev.bureau@gmail.com

Abstract. This paper proposes an approach to reducing significantly the computational complexity of optimization problems in the design of integrated rating mechanisms (IRMs). The background concepts are introduced. The representability of a given discrete function as some IRM is proved. The decomposition procedure for a particular training example on some partition of input parameters is considered, and the following results are established under some restrictive conditions. First, an IRM matrix for a particular example of an input data set can be designed by maximizing a certain polynomial. Second, a set of given examples can be implemented by some IRM matrix. Third, an IRM can be implemented on a training data set in a certain complete binary tree based on the decomposition method. Fourth, some discrete function is implemented through a given complete binary tree if the discrete functions represented by convolution matrices are implemented in each node of this tree. All these results are rigorously formulated and proved. An illustrative example of the decomposition procedure based on a complete binary tree on three leaves is given. We propose a method for finding IRMs that implement a given training set in the space of all possible complete binary trees based on the branch table. In addition, we describe the decomposition procedure according to the branch table for each partition of input parameters. Finally, the advantages of the proposed method are outlined.

Keywords: integrated rating mechanism, discrete function, assessment, decomposition.

INTRODUCTION

Training models based on precedents is a well-known practice [1–3] going beyond the field of machine learning. In recent years, the development of training procedures for integrated rating mechanisms (IRMs) has attracted the attention of researchers.

IRMs are widely used as multidimensional assessment and ranking systems for management and control in organizational and production systems [4–9]. When used for complex systems (e.g., organizational and production systems), the integrated rating procedure allows dealing with the typical difficulties of complex object assessment [10, 11]. The basic application of IRMs is ordinal ranking or classification with a predetermined number of classes for a finite set of multicriteria alternatives [12–14]. The main components of IRMs are a binary tree and convolution matrices, which yield a complex assessment based on the values of several input indicators. Recently, several approaches have been proposed to design (in other words, identify) convolution matrices by a particular binary tree [15, 16]. This paper introduces a design approach that further develops the method outlined in [15]. The approach under consideration is intended to settle the difficulties associated with the complexity of solving the optimization problem during IRM matrix design. For this purpose, we adopt the decomposition method of discrete functions. Also, a topical problem is finding a set of IRMs implementing a given data set.

Many researchers showed interest in the possibility of functional decomposition. For example, A.N. Kolmogorov [17] and V.I. Arnold [18] studied the decomposability of continuous functions. For the class of discrete functions, V.S. Vykhovanets [19] constructed the decomposition procedure of algebraic functions and analyzed the identification problem of a discrete system using a spectral decomposition; for example, see [20]. The complexity of the representation of Boolean functions was investigated by S.V. Yablonsky [21]. In the paper [22], A.V. Kuznetsov considered repetition-free Boolean functions. Also, we emphasize the work [23] on multicriteria assessment by V.A. Glotov and V.V. Pavel'ev; the authors described the application of decomposition to construct a criterion-target structure. V.N. Burkov and I.V. Burkova with colleagues studied the dichotomous function representation [24, 25] in terms of solving discrete optimization problems, including application to complex assessment.

1. BASIC NOTIONS AND DEFINITIONS

Consider a finite set of indicators $L \subset \mathbb{N}$, $|L| = l$, to rate some object on a discrete scale or rank several objects. For the IRM identification problem, assume that there is a finite set $K_i \subset \mathbb{N}$ of possible values of each indicator $i \in L$, where $k_i \in K_i$ is an assessment of an individual parameter. The vector $k = (k_1,..., k_l)^{\mathrm{T}}$, the set of all assessments, describes any possible state of the assessed objects. Also, there is a finite set $K_L \subset \mathbb{N}$ of possible integral values (ranks or classes) $k_L \in K_L$ for any k. Thus, we have some discrete function $f(\cdot): K_D \to K_L$. Here, $K_D = K_1 \times K_2 \times ... \times K_l$ is the (definitional) domain, with $\times$ denoting the Cartesian product of sets, and K_L is the codomain (range) of the function. This paper focuses on the discrete scales of indicators and values obtained in the nodes of a convolution tree [12]. A function f defined on a set K_D and taking values in a set K_L is a mapping of K_D into K_L such that each element x of the domain K_D is related to at most one element of the codomain K_L.

Definition 1. An IRM with a binary tree and matrix convolutions is a function $f(\cdot): K_D \to K_L$ for which indicators L are leaves of a complete binary tree, i.e., a digraph $G = (V, E)$:

- $V = L \cup \hat{L}$, where $\hat{L} = \{l+1,...,2l-1\}$.
- $E = \{e_{ij}\} \subseteq V \times V$:

  – $\forall i \in V \setminus \{2l-1\}\ \exists!\, j \in \hat{L} \setminus \{i\}\quad e_{ij} = 1,\ \forall t \in V \setminus j$ $e_{it} = 0$;

  – $\forall j \in L\ \forall i \in V\ e_{ij} = 0$;

  – $\forall j \in \hat{L}\ \exists! \{r,c\} \in V \setminus \{j\} \times V \setminus \{j\}: e_{rj} = 1, e_{cj} = 1$;

and $\forall j \in \hat{L}$ (an inner node of the tree, including its root):

- a finite set $K_j \subset \mathbb{N}$ with possible values $k_j \in K_j$, $K_{2l-1} = K_L$, and
- a convolution matrix $M_j = [m_{jrc} \in K_j]_{r \in \{0,...,|K_l - 1|\}, c \in \{0,...,|K_r - 1|\}}$, $\{r, c\} \in V \setminus \{j\} \times V \setminus \{j\} : e_{lj} = 1,\ e_{rj} = 1$, are given. ♦

Potentially, this definition can be extended to fuzzy [26] or continuous scales. For some IRM, by analogy with [15], let $M_f = \{M_j\}_{j \in L}$ denote the set of all its convolution matrices. This paper is devoted to IRMs with a single scale such that $\forall j \in V\ K_j = K_L$. For $L \subset \mathbb{N}$, we introduce the following notations: $\Gamma_2(L)$ is the set of all complete binary trees on named leaves from the indicator set L; $IRM_{L,2}$ is the set of all IRMs for any particular binary tree $G \in \Gamma_2(L)$; $IRM_{L,G} \subseteq IRM_{L,2}$ is the set of all IRMs with such a tree. According to Definition 1, a complete binary tree in this paper is understood as a tree in which each node has either none or two child nodes.

Based on the definitions given in [15], we denote by $q = (k, k_L)$ an individual training example consisting of the assessments for each indicator and the integrated rating for a given set of indicator values and by $Q \subset K_D \otimes K_L$ a training set of the provided examples. A training set is compatible if $\forall \{q, \tilde{q}\} \subseteq Q\quad k \neq \tilde{k}$. A training set is complete if $\forall k \in K_D\quad \exists q \in Q$: $q = (k, k_L)$. A training set is given in a single scale if $\forall\, i \in L\ K_i = K_L$. For arbitrary elements $\{k, \tilde{k}\} \subseteq K_D$, the relation $k \succ \tilde{k}$ means $\forall i \in L\quad \tilde{k}_i \leq k_i$. For an arbitrary set $Q \subset K_D \otimes K_L$, we present key notions concerning the identification problem. First of all, we formalize the implementability problem of a training set.

Definition 2. A function $f(\cdot) \in IRM_{L,2}$ implements a set Q if and only if $\forall\, q \in Q\ f(k) = k_L$. ♦

We introduce the following notations: $IRM_{L,2}(Q)$ is the set of all IRMs implementing a set Q; $IRM_{L,G}(Q)$ is the set of all IRMs that implement Q and are based on a binary tree $G \in \Gamma_2(L)$. If $IRM_{L,2}(Q) \neq \emptyset$, then the set is implementable based on an IRM; if $IRM_{L,G}(Q) \neq \emptyset$, then the set Q is implementable based on an IRM with a structure G. Definition 2 can be narrowed to one particular training example: a function $f(\cdot) \in IRM_{L,2}$ implements some example $q \in Q$ if and only if $f(k) = k_L$; the sets $IRM_{L,2}(q)$ and $IRM_{L,G}(q)$ are defined by analogy.

For some finite set $K \subset \mathbb{N}$, we write its normalized representation: $\bar{K} = \{0,...,s\}$, $s = |K| - 1$. Then $\forall x \in \bar{K}$ the unitary representation is given by $\tilde{x} = (\underbrace{0,...,0}_{x},1,\underbrace{0,...,0}_{s-x})^{\mathrm{T}}$. This paper deals with IRMs with a single scale. Then for any pair $\{x, y\} \in \{0,...,\bar{K}\}^2$, where x and y are the chosen matrix column and row, respectively, the convolution result with a matrix $M = [m_{rc} \in K]_{\{r,c\} \in \bar{K}^2}$ is described by the matrix equation

$\tilde{y}^{\mathrm{T}} M\tilde{x}$. Below, we also adopt the so-called quadratic representation $(M\tilde{x}, \tilde{y})$, simplifying it to $M\tilde{x}\tilde{y}$.

2. DISCRETE FUNCTION DECOMPOSITION

The paper [15] proposed an approach to identifying IRMs with a training mechanism on discrete data. It involves a mechanism for constructing an optimization functional based on input data and a complete binary tree. This approach to the identification problem causes difficulties when solving the optimization problem with large-dimension input data. The degree of the optimization polynomial grows linearly with the number of input parameters, and the number of general constraints of the optimization problem grows exponentially: $Cn = \kappa^{l-2} ex\_num$, where Cn is the number of general constraints; $\kappa = |K|$ is the indicator value scale for the parameters in a single scale k_L; *ex_num* is the number of examples in the set Q; finally, l is the number of parameters.

As an alternative, the identification problem can be solved in steps using the separating decomposition of the function $f(X) = \Sigma(X_1, a(X_2))$, where Σ and a are some functions and $X_1 \cap X_2 = \emptyset$ are the subsets resulting from the separation of the set X; for details, see [19]. The decomposability of any continuous function of n variables into a superposition of continuous functions of fewer variables was studied by A.N. Kolmogorov and V.I. Arnold. In particular, for two variables, it was proved in [17, 18]. The paper [17] obtained the following theoretical result: any continuous function of $n \geq 3$ variables can be represented as a superposition of some continuous functions. V.A. Glotov and V.V. Pavel'ev [23] established the representability of a discrete function of n variables in the binary (separating) form. In the case of IRMs, it is easy to show (see the Appendix) the decomposability of a given function in a complete binary tree under unfixed value scales $k_i \in K_i$ $\forall i \in \{1,..., l-1\}$: due to the finiteness of k_L, the value scale of functions decomposing a given discrete function is also finite. Thus, identification problems are sequentially formulated and solved for the decomposition procedure in each node of the tree from the set $\Gamma_2(L)$.

Following [27], let the indicator decomposition structure of an arbitrary tree $G \in \Gamma_2(L)$ be denoted by $\Lambda(G) = \{L_i\}_{i\in\{1,..,l-1\}}$, where $\forall i \in \{1,...,l-1\}$ $L_i \subseteq L$ is the set of leaves (indicators) of a subtree with root node i. Then $L_1 = L$ and $\forall i \in \{1,...,l-1\}$ $|L_i| \geq 2$. Consider a given complete set Q and a given tree $G \in \Gamma_2(L)$. For any set $L_i \in \Lambda(G)$ such that $L_i \geq 2$ and its subgroup $\{L_{ir}; L_{ic}\} \subset \Lambda(G)$: $L_{ir} \cup L_{ic} = L_i$. Consider tuples $k_{(L_{ir})}, \tilde{k}_{(L_{ir})}$ and $k_{(L_{ic})}, \tilde{k}_{(L_{ic})}$ of any admissible indicator values from the sets L_{ir} and L_{ic}, respectively. For some subset of indicators $\tilde{L} \subseteq L$, we denote by $\lambda = \left(k_{(\tilde{L})}, k_{(L\setminus\tilde{L})}\right)$ the partition of the tuple of indicators k of some training example $q = (k, k_L)$ into two tuples. Due to its complete binary structure, each tree from $G \in \Gamma_2(L)$ can be assigned the set of indicators L_i; then $\lambda_i = \left(k_{(L_{ir})}, k_{(L_{ic})}\right)$ is the partition of the tuple of indicators in node i of the tree G. In each tree node, the component functions decomposing the discrete function $\varphi_i(k_{(L_i)})$ will be named in accordance with the partition and the numbering of matrices located in the tree nodes implementing the component functions of $\varphi_i(k_{(L_i)})$. For example, consider the partition $\lambda = \left(k_{(1,2)}, k_{(3,4)}\right)$; the components of the two subfunctions are named as follows: $\varphi_{i+1}\_k_1k_2$ and $\varphi_{i+2}\_k_3k_4$. For convenience, we may also name the components of the discrete function $\varphi_i(k_{(L_i)})$ in accordance with the partition λ_i as $\varphi_r(k_{(L_{ir})})$ and $\varphi_c(k_{(L_{ic})})$. If the partition consists of an individual leaf and a group, i.e., $\lambda = \left(k_{(1)}, k_{(3,4)}\right)$, we encode only the components $\varphi_{i+1}\_k_3k_4$, where $i \in \{1,..., l-1\}$. Consider an illustrative example for the proposed approach.

Example 1. Consider the case $|L| = 3$, $|K_L| = 2$ and the training example $q = ((0, 0, 0), 0)$. The unitary representation is $\tilde{q} = \left(\left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\right), \begin{pmatrix}1\\0\end{pmatrix}\right)$.

First, we analyze the implementability of some discrete function $\varphi_1(\tilde{k}_1, \tilde{k}_2, \tilde{k}_3)$. Let the decomposition functions be named using the partition $\lambda_1 = \left(k_{(1)}, k_{(2,3)}\right)$

$$\left(\begin{pmatrix}\varphi^0_{2\_00}\\ \varphi^1_{2\_00}\end{pmatrix}^{\mathrm{T}} \begin{bmatrix} \begin{pmatrix} m1^0_{00} \\ m1^1_{00}\end{pmatrix} & \begin{pmatrix} m1^0_{01} \\ m1^1_{01}\end{pmatrix} \\ \begin{pmatrix} m1^0_{10} \\ m1^1_{10}\end{pmatrix} & \begin{pmatrix} m1^0_{11} \\ m1^1_{11}\end{pmatrix}\end{bmatrix} \begin{pmatrix}1\\0\end{pmatrix}\right) = \begin{pmatrix}1\\0\end{pmatrix}$$

with the following unitary conditions: $\forall\{i, j\} \in \{0, 1\}^2$ $m1^0_{ij} + m1^1_{ij} = 1$, $m1^t_{ij} \in \{0,1\}$, $\varphi^0_{2\_00} + \varphi^1_{2\_00} = 1$, $\varphi^t_{2\_00} \in \{0, 1\}$, $\forall t \in \{0,1\}$. Simple transformations lead to the system

$$\varphi^0_{2\_00} m1^0_{00} + \varphi^1_{2\_00} m1^0_{10} = 1, \tag{1}$$

$$\varphi^0_{2\_00} m1^1_{00} + \varphi^1_{2\_00} m1^1_{10} = 0. \tag{2}$$

System (1), (2) is easily solved using binarity constraints, e.g., $\varphi^0_{2\_00} = 1$ and $m1^0_{00} = 1$. We have a part of the optimization problem corresponding to the example *q* = ((0, 0, 0), 0). ♦

Let $P(\lambda_i, q)$ denote the function on the left-hand side of equation (1). Due to the unitary approach, there exists a unique such function for any example *q*. In addition, Q_i is the data set obtained from the original one *Q* by selecting only the columns where the function $\varphi_i(k_{(L_i)})$ is defined.

Proposition 1. *For any sets* $L \subset \mathbb{N}$ *and* $K \subset \mathbb{N}$ *and any possible example q in a single scale, there exists a homogeneous polynomial* $P(\lambda_i, q)$ *of a degree not exceeding* 3 *that can be represented as the sum of* κ^{φ_num} *unique components*:

$$P(\lambda_i, q) = \sum_{j=1}^{\kappa^{\varphi\_num}} p_j \,,\ \forall j \in \{1,..., \kappa^{\varphi\_num}\}\,,$$

$$p_j = m_j \prod_{d=1}^{\varphi\_num} \varphi_d \quad , \forall d \in \{1,..., \varphi\_num\}\,.$$

Here, the notations are the following: φ_d *is the function components decomposing the function* $\varphi_i(k_{(L_i)})$; m_j *is one tuple component in some cell of the unitary encoded matrix* $\tilde{M}_i$; *q is an example from* Q_i; φ_num = 1 *when a branch and a leaf are connected to the matrix;* φ_num = 2 *when a branch pair is connected to the matrix;*

$P(\lambda_i, q) \in \{0, 1\}$;

$\varphi_i(k_{(L_i)})$ *implements q* ⇔ $P(\lambda_i, q)$ = 1.

The proof of Proposition 1 is given in the Appendix.

For the function $\varphi_i(k_{(L_i)})$, at each step of the decomposition procedure, we form an appropriate set Q_i from the set *Q* based on the leaves corresponding to the partition λ_i.

Example 2. Within the conditions of Example 1, we add another training example q_2 = ((0, 1, 0), 1) based on the same partition $\lambda_1 = \left(k_{(1)}, k_{(2,3)}\right)$. Its unitary representation is $q_2 = \left(\left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\right), \begin{pmatrix}1\\0\end{pmatrix}\right)$. Then we have the set of operations

$$\left(\begin{pmatrix}\varphi^0_{2\_00}\\ \varphi^1_{2\_00}\end{pmatrix}^{\mathrm{T}} \begin{bmatrix}\begin{pmatrix}m1^0_{00}\\ m1^1_{00}\end{pmatrix} & \begin{pmatrix}m1^0_{01}\\ m1^1_{01}\end{pmatrix}\\ \begin{pmatrix}m1^0_{10}\\ m1^1_{10}\end{pmatrix} & \begin{pmatrix}m1^0_{11}\\ m1^1_{11}\end{pmatrix}\end{bmatrix} \begin{pmatrix}1\\0\end{pmatrix}\right) = \begin{pmatrix}1\\0\end{pmatrix},$$

$$\left(\begin{pmatrix}\varphi^0_{2\_00}\\ \varphi^1_{2\_00}\end{pmatrix}^{\mathrm{T}} \begin{bmatrix}\begin{pmatrix}m1^0_{00}\\ m1^1_{00}\end{pmatrix} & \begin{pmatrix}m1^0_{01}\\ m1^1_{01}\end{pmatrix}\\ \begin{pmatrix}m1^0_{10}\\ m1^1_{10}\end{pmatrix} & \begin{pmatrix}m1^0_{11}\\ m1^1_{11}\end{pmatrix}\end{bmatrix} \begin{pmatrix}1\\0\end{pmatrix}\right) = \begin{pmatrix}0\\1\end{pmatrix}$$

with the following unitary conditions: $\forall\{i, j\} \in \{0,1\}^2$ $m1^0_{ij} + m1^1_{ij} = 1$, $m1^t_{ij} \in \{0,1\}$, $\varphi^0_{2\_00} + \varphi^1_{2\_00} = 1$. A comparison of the equations written for the first and second training examples shows that the values φ_{2_00} and φ_{2_01} must be compatible. Therefore, when constructing the function $\varphi_i(k_{(L_i)})$ implementing the examples q_1 and q_2 simultaneously, we require $\varphi_{2\_00}{}^{\mathrm{T}}\varphi_{2\_01} = 0$. ♦

Corollary 1. *For any sets* $L \subset \mathbb{N}$ *and* $K \subset \mathbb{N}$ *and any possible* $Q_i \subset K^{l+1}$ *with a single scale, the discrete function* $\varphi_i(k_{(L_i)})$ *implements the set* Q_i *if* $\sum_{q\in Q_i} P(\lambda_i, q) = |Q_i|$ *for any possible example q in the single scale considering the compatibility of the functions* $\varphi_r(k_{(L_{ir})})$ *and* $\varphi_c(k_{(L_{ic})})$ *for all examples in the set* Q_i.

The proof of Corollary 1 is given in the Appendix.

If two branches join the matrix under consideration, then φ_num = 2 and the compatibility of the functions $\varphi_r(k_{(L_{ir})})$ and $\varphi_c(k_{(L_{ic})})$ should be verified for each branch.

Proposition 2. *For any sets* $L \subset \mathbb{N}$ *and* $K \subset \mathbb{N}$, *the mechanism* $IRM_{G,2}(q)$ *is represented as a decomposition of the function* $f(\tilde{k}_1,..., \tilde{k}_l)$ *and implements a set* $Q \subset K^{l+1}$ *if the function* $\varphi_i(k_{(L_i)})$ *for some partition sequence* $\Lambda(G)$ *corresponding to the tree G implements the set* Q_i, *for i* ∈ {1,..., $l-1$}.

Corollary 2. *If* $\sum_{q\in Q_i} P(\lambda_i, q) < |Q_i|$, ∀ *i* ∈ {1,..., $l-1$}, *then* $IRM_{G,2}(Q)$ = Ø.

The proofs of Proposition 2 and Corollary 2 are given in the Appendix.

Thus, if the optimization problem $\operatorname*{Arg\,max}_{m_{rc}, \varphi_r, \varphi_c} \sum_{q\in Q_1} P(\lambda_1, q)$ has a solution such that $\sum_{q\in Q_1} P(\lambda_1, q) = |Q_1|$, we can continue the decomposition procedure to the next tree node with the found components of the discrete function $\varphi_1(k_{(L_1)})$. On this way, $\varphi_1(k_{(L_1)})$ and the found values of the matrix M_1 can be used for the decomposition procedure on the subtrees $\{L_{1r}; L_{1c}\} \subset \Lambda(G)$ on the indicator values $k_{(L_{1r})}$ and $k_{(L_{1c})}$, respectively. This approach sequentially yields the vectors $\varphi_i(k_{(L_i)})$ and the matrix M_i, $\forall\, i \in \{1,..., l-1\}$ for the structure $\Lambda(G)$.

Example 3. Consider the initial data below.

Table 1

Initial data for the decomposition procedure

q	k_1	k_2	k_3	k_4	k_L
1	0	0	0	0	0
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	1	1	0	1
6	0	1	0	1	1
7	1	0	1	1	0
8	1	1	1	1	0

First, we analyze the implementability of the function $f=\varphi_1(\tilde{k}_1,\, \varphi_2(\tilde{k}_2,\, \tilde{k}_3,\, \tilde{k}_4))$; if $\varphi_1(k_{(L_1)})$ is available in the given scale k_L, we proceed to the implementation of the discrete function $\varphi_2(k_{(L_2)})$. Let the first-step partition be $\lambda_1=\left(k_{(1)},\, k_{(2,3,4)}\right)$. The equations for the first example of the first step have the form

$$\left(\begin{pmatrix}\varphi^0_{2\_000}\\ \varphi^1_{2\_000}\end{pmatrix}^{\mathrm T}\begin{bmatrix}\begin{pmatrix}m1^0_{00}\\ m1^1_{00}\end{pmatrix} & \begin{pmatrix}m1^0_{01}\\ m1^1_{01}\end{pmatrix}\\ \begin{pmatrix}m1^0_{10}\\ m1^1_{10}\end{pmatrix} & \begin{pmatrix}m1^0_{11}\\ m1^1_{11}\end{pmatrix}\end{bmatrix}\begin{pmatrix}1\\0\end{pmatrix}\right)=\begin{pmatrix}1\\0\end{pmatrix}$$

with the following unitary conditions: $\forall\{i,j\}\in\{0,1\}^2$ $m1^0_{ij}+m1^1_{ij}=1,\ m1^t_{ij}\in\{0,1\}$, $\varphi^0_{2\_000}+\varphi^1_{2\_000}=1$, $\varphi^t_{2\_000}\in\{0,1\}$, $\forall t\in\{0,1\}$:

$$\varphi^0_{2\_000}m1^0_{00}+\varphi^1_{2\_000}m1^0_{10}=1,$$
$$\varphi^0_{2\_000}m1^1_{00}+\varphi^1_{2\_000}m1^1_{10}=0.$$

Next, using the scheme described in Example 1, we derive equations for all examples:

$$\varphi^0_{2\_000}m1^0_{00}+\varphi^1_{2\_000}m1^0_{10}=1;\ \varphi^0_{2\_100}m1^1_{00}+\varphi^1_{2\_100}m1^1_{10}=1;$$
$$\varphi^0_{2\_100}m1^0_{01}+\varphi^1_{2\_100}m1^0_{11}=1;\ \varphi^0_{2\_010}m1^1_{00}+\varphi^1_{2\_010}m1^1_{10}=1;$$
$$\varphi^0_{2\_110}m1^1_{10}+\varphi^1_{2\_110}m1^1_{11}=1;\ \varphi^0_{2\_101}m1^1_{00}+\varphi^1_{2\_101}m1^1_{10}=1;$$
$$\varphi^0_{2\_011}m1^0_{01}+\varphi^1_{2\_011}m1^0_{11}=1;\ \varphi^0_{2\_111}m1^0_{01}+\varphi^1_{2\_111}m1^0_{11}=1.$$

The corresponding optimization problem is

$$\begin{aligned}&\varphi^0_{2\_000}m1^0_{00}+\varphi^1_{2\_000}m1^0_{10}+\varphi^0_{2\_100}m1^1_{00}\\&+\varphi^1_{2\_100}m1^1_{10}+\varphi^0_{2\_100}m1^0_{01}\\&+\varphi^1_{2\_100}m1^0_{11}+\varphi^0_{2\_010}m1^1_{00}+\varphi^1_{2\_010}m1^1_{10}+\varphi^0_{2\_110}m1^1_{10}\\&+\varphi^1_{2\_110}m1^1_{11}+\varphi^0_{2\_101}m1^1_{00}+\varphi^1_{2\_101}m1^1_{10}+\varphi^0_{2\_011}m1^0_{01}\\&+\varphi^1_{2\_011}m1^0_{11}+\varphi^0_{2\_111}m1^0_{01}+\varphi^1_{2\_111}m1^0_{11}\to\max.\end{aligned}\tag{3}$$

In addition, there are constraints due to conflicts between equations for different steps. For example, from the expression

$$\left(\begin{pmatrix}\varphi^0_{2\_000}\\ \varphi^1_{2\_000}\end{pmatrix}^{\mathrm T}\begin{bmatrix}\begin{pmatrix}m1^0_{00}\\ m1^1_{00}\end{pmatrix} & \begin{pmatrix}m1^0_{01}\\ m1^1_{01}\end{pmatrix}\\ \begin{pmatrix}m1^0_{10}\\ m1^1_{10}\end{pmatrix} & \begin{pmatrix}m1^0_{11}\\ m1^1_{11}\end{pmatrix}\end{bmatrix}\begin{pmatrix}1\\0\end{pmatrix}\right)=\begin{pmatrix}1\\0\end{pmatrix}$$

for the first example and

$$\left(\begin{pmatrix}\varphi^0_{2\_100}\\ \varphi^1_{2\_100}\end{pmatrix}^{\mathrm T}\begin{bmatrix}\begin{pmatrix}m1^0_{00}\\ m1^1_{00}\end{pmatrix} & \begin{pmatrix}m1^0_{01}\\ m1^1_{01}\end{pmatrix}\\ \begin{pmatrix}m1^0_{10}\\ m1^1_{10}\end{pmatrix} & \begin{pmatrix}m1^0_{11}\\ m1^1_{11}\end{pmatrix}\end{bmatrix}\begin{pmatrix}1\\0\end{pmatrix}\right)=\begin{pmatrix}0\\1\end{pmatrix}$$

for the second one, we write $\begin{pmatrix}\varphi^0_{2\_000}\\ \varphi^1_{2\_000}\end{pmatrix}^{\mathrm T}\begin{pmatrix}\varphi^0_{2\_100}\\ \varphi^1_{2\_100}\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$ or $\varphi_{2\_000}\varphi_{2\_100}=0$; $\varphi_{2\_000}\varphi_{2\_010}=0$; $\varphi_{2\_110}\varphi_{2\_100}=0$; $\varphi_{2\_000}\varphi_{2\_101}=0$; $\varphi_{2\_110}\varphi_{2\_011}=0$; $\varphi_{2\_110}\varphi_{2\_111}=0$.

The solution of problem (3) is the matrix $\tilde{M}_1=\begin{bmatrix}\begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix}\\ \begin{pmatrix}0\\1\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix}\end{bmatrix}$ and the vector $\tilde{\varphi}_2=\left(\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix}\right)$. Based on these values of the discrete function $\varphi_2(k_{(L_2)})$, we compile a data table for the second step of the decomposition procedure of the function $f=\varphi_2(\tilde{k}_3,\, \varphi_3(\tilde{k}_1,\, \tilde{k}_2))$, placing the values of the function $\varphi_2(k_{(L_2)})$ in column $\tilde{k}_L$; see Table 2.

Table 2

Data on the second step of the decomposition procedure

q	k_2	k_3	k_4	k_L
1	0	0	0	0
2	1	0	0	1
3	0	1	0	1
4	1	1	0	0
5	1	0	1	1
6	0	1	1	1
7	1	1	1	1

Let the second-step partition be $\lambda_2=\left(k_{(4)},k_{(2,3)}\right)$. The equation for this partition has the form

$$\left(\begin{pmatrix}\varphi_{3\_00}^0\\ \varphi_{3\_00}^1\end{pmatrix}^{\mathrm{T}}\begin{bmatrix}\begin{pmatrix}m2_{00}^0\\ m2_{00}^1\end{pmatrix} & \begin{pmatrix}m2_{01}^0\\ m2_{01}^1\end{pmatrix}\\ \begin{pmatrix}m2_{10}^0\\ m2_{10}^1\end{pmatrix} & \begin{pmatrix}m2_{11}^0\\ m2_{11}^1\end{pmatrix}\end{bmatrix}\begin{pmatrix}1\\ 0\end{pmatrix}\right)=\begin{pmatrix}1\\ 0\end{pmatrix}.$$

with the following unitary conditions: $\forall\{i,\ j\}\in\{0,1\}^2$ $m2_{ij}^0+m2_{ij}^1=1$, $m2_{ij}^t\in\{0,1\}$, $\varphi_{3\_00}^0+\varphi_{3\_00}^1=1$, $\varphi_{3\_00}^t\in\{0,1\}$, $\forall t\in\{0,1\}$:

$$\varphi_{3\_00}^0 m2_{00}^0+\varphi_{3\_00}^1 m2_{10}^0=1,$$

$$\varphi_{3\_00}^0 b_{00}^1+\varphi_{3\_00}^1 b_{10}^1=0.$$

The resulting set of equations is

$$\varphi_{3\_00}^0 m2_{00}^0+\varphi_{3\_00}^1 m2_{10}^0=1;\ \varphi_{3\_10}^0 m2_{00}^1+\varphi_{3\_10}^1 m2_{10}^1=1;$$

$$\varphi_{3\_01}^0 m2_{00}^1+\varphi_{3\_01}^1 m2_{10}^1=1;\ \varphi_{3\_11}^0 m2_{00}^0+\varphi_{3\_11}^1 m2_{10}^0=1;$$

$$\varphi_{3\_10}^0 m2_{10}^1+\varphi_{3\_10}^1 m2_{11}^1=1;\ \varphi_{3\_01}^0 m2_{10}^1+\varphi_{3\_01}^1 m2_{11}^1=1;$$

$$\varphi_{3\_11}^0 m2_{01}^1+\varphi_{3\_11}^1 m2_{11}^1=1.$$

The corresponding optimization problem has the form

$$\begin{aligned}&\varphi_{3\_00}^0 m2_{00}^0+\varphi_{3\_00}^1 m2_{10}^0+\varphi_{3\_10}^0 m2_{00}^1\\&+\varphi_{3\_10}^1 m2_{10}^1+\varphi_{3\_01}^0 m2_{00}^1+\varphi_{3\_01}^1 m2_{10}^1\\&+\varphi_{3\_11}^0 m2_{00}^0+\varphi_{3\_11}^1 m2_{10}^0+\varphi_{3\_10}^0 m2_{10}^1\\&+\varphi_{3\_10}^1 m2_{11}^1+\varphi_{3\_01}^0 m2_{10}^1+\varphi_{3\_01}^1 m2_{11}^1\\&+\varphi_{3\_11}^0 m2_{01}^1+\varphi_{3\_11}^1 m2_{11}^1\to\max.\end{aligned}\tag{4}$$

In addition, the analysis of conflicts between equations for different examples gives the constraints

$$\varphi_{3\_10}\varphi_{3\_00}=0;\ \varphi_{3\_01}\varphi_{3\_00}=0;\ \varphi_{3\_00}=\varphi_{3\_11}.$$

The solution of problem (4) is the matrix $\tilde{M}_2=\begin{bmatrix}\begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix}\\ \begin{pmatrix}0\\1\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix}\end{bmatrix}$ and the vector $\tilde{\varphi}_3=\left(\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}1\\0\end{pmatrix}\right)$.

Obviously, the discrete function $\varphi_3(\tilde{k}_1,\tilde{k}_2)$ needs no further decomposition. The result is $\tilde{M}_3=\begin{bmatrix}\begin{pmatrix}1\\0\end{pmatrix} & \begin{pmatrix}0\\1\end{pmatrix}\\ \begin{pmatrix}0\\1\end{pmatrix} & \begin{pmatrix}1\\0\end{pmatrix}\end{bmatrix}$. ♦

3. BRANCH TABLE

When finding all structures on l leaves from the set $\Gamma_2(L)$, we should check equivalence groups; see the corresponding mechanism in [27]. Based on the analysis results, the number of partitions at each step can be reduced by eliminating leaf combinations non-implementable in a given scale k_L. It is convenient to summarize the results in a branch table. This is a compact step-by-step representation of leaf combinations for implementability analysis within the decomposition procedure.

If some leaf combinations are admissible by the analysis results of equivalence groups, we compile the branch table starting from the groups with two leaves, $|L_i|=2$. Taking only the admissible leaf combinations reduces the number of structures considered. Thus, we list the checked groups of three leaves. If there are no two-leaf groups among the admissible ones, the further procedure becomes pointless: it means that no terminal matrix (a matrix taking values of two leaves) can be designed within the given scale.

The admissible groups are placed in the branch table. The admissible groups consisting of two and individual leaves form combinations of three leaves, $|L_i|=3$. The resulting branches are placed in the table column corresponding to their group name. Next, the admissible groups consisting of three, two, and individual leaves form combinations with $|L_i|=4$, yielding branches of four leaves $|L_i|=4$. Note that starting from $|L_i|=4$, we consider the branches included in the equivalence group analysis results and placed in the branch table. In other words, the branches with $|L_i|>2$ are designed from the admissible branches with fewer leaves. The resulting branches are placed in the table column corresponding to their group name. The procedure continues until reaching the tree root, $|L_i|=l$. As a result, any tree $G\in\Gamma_2(L)$ whose structure $\Lambda(G)$ belongs to the list of admissible leaf groups must be considered in the IRM identification problem for the training set Q.

A certain tree $G\in\Gamma_2(L)$ consisting of admissible subbranches is considered using the branch table as follows. The partitions λ_i are sequentially taken from the table column with the largest number. The optimization problem $\operatorname{Arg}\max\limits_{m_{rc},\varphi_r,\varphi_c}\sum\limits_{q\in Q_1}P(\lambda_i,q)$ is constructed for the set Q and each partition λ_i. If it has a solution within the admissible scale, the matrix M_i is saved, and the resulting values of the functions $\varphi_r(k_{(L_{ir})})$ and $\varphi_c(k_{(L_{ic})})$ are used to find a solution for the subbranches of the partitions λ_i. After examining all the partitions λ_i in the table column with the largest number, the consideration proceeds to the column with the lower number.

This approach has the following advantage: if the problem $\operatorname{Arg}\max\limits_{m_{rc},\varphi_r,\varphi_c}\sum\limits_{q\in Q_1}P(\lambda_i,q)$ is unsolvable, we can exclude from further consideration the entire family of subbranches generated by the partition λ_i. For example, if there is no solution in the scale k_L for $\lambda_1=\left(k_{(1)},\,k_{(2.3.4)}\right)$ (see Table 3), we can exclude the

Table 3

An example of the branch table

#	0	1	2	3	4	5
L_i	1 2	2 3	3 4	1 2 4	2 3 4	1 2 3 4
λ_i	1 2	2 3	3 4	4, 1 2	4, 2 3 2, 3 4	3, 1 2 4 1, 2 3 4 1 2, 3 4

structures *M*1*l*1*M*2*l*4*M*3*l*2*l*3 and *M*1*l*1*M*2*l*2*M*3*l*3*l*4, neglecting decompositions $\lambda_2 = \left(k_{(3)},\, k_{(1,2)}\right)$ and $\lambda_2 = \left(k_{(1)}, k_{(2,3)}\right)$. For details, we refer to the paper [27].

CONCLUSIONS

This paper has considered an approach to designing integrated rating mechanisms based on separating decomposition. The branch table has been proposed as a decomposition scheme. In contrast to the approach described in [15], optimization problems are sequentially constructed and solved for each IRM node; according to Proposition 1, the optimization polynomial does not depend on the number of input parameters. In each node of the complete binary tree, the degree of the polynomial does not exceed 3. Due to these properties, the optimization problems are quickly solved by an optimizer. For example, Gurobi 9.5.0 [28] solves the optimization problem with 8 quadratic constraints and 16 general constraints (the first step of the second example) in 30 ms on a PC with AMD Ryzen 7 4800H processor and 16GB RAM. The proposed approach has another advantage as follows: if some step of the decomposition procedure of a discrete function yields no solutions in a given scale, this procedure becomes pointless to continue (the problem will be unsolvable in this scale). Further research will deal with sorting the most promising solutions from the general solution pool for a given function.

APPENDIX

P r o o f (the separating decomposability of some discrete function of *n* variables under unfixed parameter value scales).

As mentioned above, in some complete binary tree, an IRM is defined through a set of convolution matrices $M_f = \{M_j\}_{j\in L}$. In the IRM structure, the convolution operation $\tilde{M}_i \tilde{x}_i \tilde{y}_j$ is performed using each matrix M_i. Each individual matrix M_i implements some discrete function $\varphi_i(k_{(L_i)})$. For an arbitrary tree $G \in \Gamma_2(L)$, the indicator decomposition structure consists of $\Lambda(G) = \{L_i\}_{i\in\{1,\dots,l-1\}}$ so that $\forall i \in \{1,\dots,l-1\}$ $L_i \subseteq L$ is the set of leaves (indicators) of a subtree with the root node i. The set L_i has some subgroups $\{L_{ir}; L_{ic}\} \subset \Lambda(G) : L_{ir} \cup L_{ic} = L_i$. In general, the dimension of the matrix $\tilde{M}_i$ required to implement the discrete function $\varphi_i(k_{(L_i)})$ on the partition $\lambda_i = \left(k_{(L_{ir})}, k_{(L_{ic})}\right)$ is unknown. Based on the description of equivalence groups [27], the number of equivalence groups for the partition λ_i, and hence the corresponding dimension of the matrix $\tilde{M}_i$, cannot exceed the number of combinations encoded by the indicators of the subsets $K_{(L_{ir})} = \prod_{j\in L_{ir}} K_j$ and $K_{(L_{ic})} = \prod_{j\in L_{ic}} K_j$ for the subsets of row and column indicators, respectively. It may be necessary to place examples in the matrix M_i on different cells, each corresponding to a different equivalence group of the current decomposition step. That is, in the worst case, an individual matrix cell should be provided for each training example. In other words, any subset of variables for a discrete function contains a finite number of combinations of their values; this number is an estimate of the maximum number of matrix rows or columns. Consequently, the dimension of some discrete function $\varphi_i(k_{(L_i)})$ will not exceed $K_{(L_i)}$, $L_{ir} \cup L_{ic} = L_i$. Thus, at each decomposition step of the function *f*, the dimension of the matrix $\tilde{M}_i$, $i \in l-1$, is sufficient to implement the function $\varphi_i(k_{(L_i)})$ by construction. This matrix design approach ensures that the IRM constructed from any admissible complete binary tree will implement the given function *f*. ♦

P r o o f of Proposition 1.

Due to the unitary notation, the operation within an IRM with an individual set of indicator values is represented as some stepwise function $f(\tilde{k}_0,\dots,\tilde{k}_{l-1})$ decomposable in some set of partitions $\Lambda(G)$. At each decomposition step (at each node of the tree *G*), the discrete function $\varphi_i(k_{(L_i)})$ is defined on a subset of leaves L_i in the partition λ_i based on an example set Q_i constructed from the set *Q* by selecting leaves of the corresponding subsets L_i. Therefore, the dimension $k_{(L_i)}$ is determined based on the subset L_i where the function $\varphi_i(k_{(L_i)})$ is defined. The function $\varphi_i(k_{(L_i)})$ is a matrix operation on an individual matrix, $\tilde{y}^{\mathrm{T}}\tilde{M}\tilde{x}$, where the vectors $\tilde{x}$ and $\tilde{y}$ have dimension κ and the matrix $\tilde{M}$ have dimension $\kappa\times\kappa$. Obviously, each such operation yields a vector of dimension $\kappa \in \mathbb{N}$ containing the components of an individual matrix cell. An example of this operation on a matrix M_i is as follows:

$$\begin{pmatrix} y^0(q^i) \\ y^1(q^i) \end{pmatrix}^{\mathrm{T}} \begin{bmatrix} \begin{pmatrix} m1^0_{00} \\ m1^1_{00} \end{pmatrix} & \begin{pmatrix} m1^0_{01} \\ m1^1_{01} \end{pmatrix} \\ \begin{pmatrix} m1^0_{10} \\ m1^1_{10} \end{pmatrix} & \begin{pmatrix} m1^0_{11} \\ m1^1_{11} \end{pmatrix} \end{bmatrix} \begin{pmatrix} x^0(q^i) \\ x^1(q^i) \end{pmatrix}.$$

That is, the final result of the operation $\tilde{y}^{\mathrm{T}}\tilde{M}\tilde{x}$ will also be a vector of dimension κ. Each component of this vector will be represented by a homogeneous polynomial of degree 3 since two branches join the matrix at each step. The vectors $\tilde{x}$ and $\tilde{y}$ can be both some functions (the components of $\varphi_i(k_{(L_i)})$) and leaves, and the values of all leaves are given by unitary vectors. Hence, the final degree of the polynomial will not exceed 3, retaining only the terms not multiplied by the zero components of the leaf vector. Each polynomial term will have the form $m_j \prod_{d=1}^{\varphi\_num} \varphi_d$, where m_j is one component of the tuple in some cell of the unitary encoded matrix $\tilde{M}$. The uniqueness of each term also follows from the essence of the described operation.

The cells of all matrices must contain unitary vectors. Therefore, each component of the vector defined by the operation $\tilde{y}^{\mathrm{T}}\tilde{M}\tilde{x}$ can be either 0 or 1. With the scheme $\tilde{y}^{\mathrm{T}}\tilde{M}\tilde{x}$ and the given values of the function *f*, we obtain the equations

$$\begin{pmatrix} y^0(q^i) \\ y^1(q^i) \end{pmatrix}^{\mathrm{T}} \begin{bmatrix} \begin{pmatrix} m1^0_{00} \\ m1^1_{00} \end{pmatrix} & \begin{pmatrix} m1^0_{01} \\ m1^1_{01} \end{pmatrix} \\ \begin{pmatrix} m1^0_{10} \\ m1^1_{10} \end{pmatrix} & \begin{pmatrix} m1^0_{11} \\ m1^1_{11} \end{pmatrix} \end{bmatrix} \begin{pmatrix} x^0(q^i) \\ x^1(q^i) \end{pmatrix} = \begin{pmatrix} K^0(q^i) \\ K^1(q^i) \end{pmatrix}.$$

That the function $\varphi_i(k_{(L_i)})$ implements a single example *q* from the set Q_i actually means $\tilde{y}^{\mathrm{T}}\tilde{M}_i\tilde{x} = \varphi_i(q)$. Since the vector $\varphi_i(q)$ is unitary, the resulting vector of the operation $\tilde{y}^{\mathrm{T}}\tilde{M}\tilde{x}$ has only one component equal to 1, the same as in the vector $\varphi_i(q)$; all others components must be 0. That is, $\varphi_i(q)^{\mathrm{T}}\tilde{y}^{\mathrm{T}}\tilde{M}\tilde{x} = 1$.

Denoting by $P(\lambda_i, q)$ the polynomial corresponding to the vector component determined by the function $\varphi_i(q)$ (must equal 1), we establish all items of Proposition 1. ♦

P r o o f of Corollary 1.

According to the proof of Proposition 1, the function $\varphi_i(k_{(L_i)})$ implements some training example $q \Leftrightarrow P(\lambda_i, q) = 1$. Hence, if all $|Q_i|$ examples are implemented, we have $\sum_{q\in Q_i} P(\lambda_i, q) = |Q_i|$. ♦

P r o o f Proposition 2.

For any $L \subset \mathbb{N}$ and $K \subset \mathbb{N}$, any set $Q \subset K^{l+1}$ according to some sequence of partitions $\Lambda(G)$ on a tree *G* from the set of complete binary trees $G_2(L)$, and the function $f(\tilde{k}_1,..., \tilde{k}_l)$ can be represented as a superposition of functions of fewer variables $\varphi_i(k_{(L_i)})$, $\forall i \in\{1,.., l-1\}$, defined on the datasets Q_i obtained from the set *Q*. Indeed, all components of the function $\varphi_i(k_{(L_i)})$, $\varphi_r(k_{(L_{ir})})$ and $\varphi_c(k_{(L_{ic})})$, are obtained from the function $\varphi_i(k_{(L_i)})$, $\forall i \in\{1,..., l-1\}$ by construction, and the values of the function $\varphi_1(k_{(L_1)})$ are defined through the set *Q*. ♦

P r o o f of Corollary 2.

By Proposition 2, the function $f(\tilde{k}_1,..., \tilde{k}_l)$ as a superposition of functions of fewer variables $\varphi_i(k_{(L_i)})$, $\forall i \in\{1,..., l-1\}$, can be constructed through the sequential optimization $\max \sum_{q\in Q_i} P(\lambda_i, q)$ with finding at each step the function $\varphi_i(k_{(L_i)})$, $\forall i \in \{1,..., l-1\}$. We have the values of the function $\varphi_1(k_{(L_1)})$ from the set *Q*; we calculate the values of the functions $\varphi_r(k_{(L_{ir})})$ and $\varphi_c(k_{(L_{ic})})$, $\forall i \in \{2,..., l-1\}$, for each decomposable function based on the function $\varphi_i(k_{(L_i)})$. If the problem is unsolvable at any step *i* for some of the $|Q_i|$ examples, it will have no solution for the components of the function $\varphi_i(k_{(L_i)})$ as well. Consequently, $IRM_{G,2}(Q) = \varnothing$. ♦

REFERENCES

1. Raschka, S., *Python Machine Learning*, Packt Publishing, 2015.
2. Knaeble, M., Nadj, M., and Maedche, A., Oracle or Teacher? A Systematic Overview of Research on Interactive Labeling for Machine Learning, in *WI2020 Zentrale Tracks*, 2020, pp. 2–16. DOI:10.30844/wi_2020_a1-knaeble
3. Simard, P., Amershi, S., Chickering, M., et al., Machine Teaching: A New Paradigm for Building Machine Learning Systems, *arXiv:1707.06742*, 2017.
4. Gorelikov, N.I., Designing a Sectoral Model for Management of Development and Manufacture of New Products, *Automation and Remote Control*, 1984, vol. 45, no. 5, pp. 598–604.
5. Burkov, V.N., Novikov, D.A., and Shchepkin, A.V., *Control Mechanisms for Ecological-Economic Systems*, Springer, 2015.
6. Korgin, N.A. and Rozhdestvenskaya, S.M., Concordant Approach for R&D Projects' Evaluation and Ranking for For-

mation of Programs for the Creation of Scientific and Technological Potential, *Proceedings of the 11th IEEE International Conference on Application of Information and Communication Technologies (AICT2017)*, Moscow, IEEE, 2017, vol. 2, pp. 358–362.
7. Shchepkin, A., Application of Integrated Mechanism in Financing Project Works, *Proceedings of the 13th International Conference on Management of Large-Scale System Development*: *(MLSD'2020)*, Moscow, 2020, pp. 1–4.
8. Zheglova, Y. and Titarenko, B., Methodology for the Integrated Assessment of Design Solutions for Foundation Pit Fences Based on the Theory of Active Systems, *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 869, art. no. 052012.
9. Burkov, V., et al., Models and Management Structure for the Development and Implementation of Innovative Technologies in Railway Transportation. I. Mechanisms of Priority Projects Selection and Resource Allocation, *Automation and Remote Control*, 2020, vol 81, no. 7, pp. 1316–1329.
10. Firsova, E.A., Firsov, S.S., and Mayorova, A.N., Evaluation of the Effectiveness of Enterprise Organizational Restructuring, *Azimuth of Scientific Research: Economics and Administration*, 2017, vol. 6, no. 2(19), pp. 283–286. (In Russian.)
11. Andronnikova, N.G., Burkov, V.N., and Leont'ev, S.V., *Kompleksnoe otsenivanie v zadachakh regional'nogo upravleniya* (Complex Assessment in Regional Management), Moscow: Trapeznikov Institute of Control Sciences RAS, 2002. (In Russian.)
12. Burkov, V.N., Gorelikov, N.I., and Cherkashin, A.M., Methodological Foundations of Integrated Assessment of Enterprises Activities Considering Their Progressiveness in VPO Soyuzelektropribor, *Pribory i Sistemy Upravleniya*, 1982, no. 11, p. 21. (In Russian.)
13. Blachev, R.N., Peculiarities of the Binary Aggregation Procedure for Multi-criterial Expert Decisions, *Automation and Remote Control*, 1997, vol. 58, no. 5, pp. 817–821. (In Russian.)
14. Mariel, P., Hoyos, D., Meyerhoff, J., et al., *Environmental Valuation with Discrete Choice Experiments: Guidance on Design, Implementation and Data Analysis*, Berlin: Springer Nature, 2021.
15. Burkov, V., Korgin, N., and Sergeev, V., Identification of Integrated Rating Mechanisms as Optimization Problem, *Proceedings of the 13th International Conference on Management of Large-Scale System Development* (*MLSD'2020*), Moscow: Trapeznikov Institute of Control Sciences RAS, 2020, pp. 1–5.
16. Alekseev, A., Identification of Integrated Rating Mechanisms Based on Training Set, *Proceedings of the 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency* (*SUMMA*), IEEE, 2020, pp. 398–403.
17. Kolmogorov, A.N., On the Representation of Continuous Functions of Several Variables by Superpositions of Continuous Functions of Fewer Variables, *Doklady Akad. Nauk SSSR*, 1956, vol. 108, no. 2. (In Russian.)
18. Arnold, V.I., On Functions of Three Variables, *Doklady Akad. Nauk SSSR*, 1957, vol. 114, no. 4, pp. 679–681. (In Russian.)
19. Vykhovanets, V.S., Algebraic Decomposition of Discrete Functions, *Automation and Remote Control*, 2006, vol. 67, no. 3, pp. 361–392.
20. Vykhovanets, V.S., Spectral Identification of Discrete Systems, *Proceedings of the 8th International Conference "System Identification and Control Problems" (SICPRO '09)*, Moscow: Trapeznikov Institute of Control Sciences RAS, 2009, pp. 1500–1517. (In Russian.)
21. Yablonskii, S.V., On Algorithmic Difficulties of Synthesizing Minimal Contact Circuits, in *Problemy kibernetiki* (Problems of Cybernetics), Moscow: Fizmatgiz, 1959, no. 2, pp. 75–121. (In Russian.)
22. Kuznetsov, A.V., Repetition-Free Contact Schemes and Repetition-Free Superpositions of Functions of Algebra of Logic, *Trudy Mat. Inst. Steklov.*, 1958, vol. 51, pp. 186–225. (In Russian.)
23. Glotov, V.A. and Pavel'ev, V.V., *Vektornaya stratifikatsiya* (Vector Stratification), Moscow: Nauka, 1984. (In Russian.)
24. Burkova, I.V., Dranko, O.I., Kryukov, S.V., and Strukov, A.Yu., Dichotomizing Representation at the Complex Estimation of the Enterprises, *Vestnik Voronezh. Gos. Tekh. Univ.*, 2010, vol. 6, no. 11, pp. 133–136. (In Russian.)
25. Burkov, V.N., Burkova, I.V., and Popok, M.V., The Method of Dychotomic Programming, *Large-Scale Systems Control*, 2004, no. 9, pp. 57–75. (In Russian.)
26. Alekseev, A.O., Kalent'eva, A.S., Vychegzhanin, A.V., and Klimets, D.V., Algorithmic Foundations of Fuzzy Integrated Assessment Procedure for Different-Nature Objects, *Fundamental Research*, 2014, no. 3-3, pp. 469–474. (In Russian.)
27. Korgin, N.A. and Sergeev, V.A., Identification of Integrated Rating Mechanisms on Complete Data Sets, in *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: Proceedings of IFIP WG 5.7 International Conference*, Berlin: Springer, 2021, vol. 630, pp. 610–616.
28. *Gurobi Optimizer Reference Manual*, Gurobi Optimization, LLC, 2020. URL: https://www.gurobi.com

This paper was recommended for publication by V.N. Burkov, a member of the Editorial Board.

Received November 23, 2022, and revised December 14, 2022.
Accepted December 26, 2022.

Author information

Sergeev, Vladimir Aleksandrovich. Researcher, Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ sergeev.bureau@gmail.com

Cite this paper

Sergeev, V.A., Design of Integrated Rating Mechanisms Based on Separating Decomposition. *Control Sciences* **6,** 2–10 (2022). http://doi.org/10.25728/cs.2022.6.1

Original Russian Text© Sergeev, V.A., 2022, published in *Problemy Upravleniya*, 2022, no. 6, pp. 3–13.

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com