Multi-Criteria Decision Discrete Model for Selecting Software Components in Component-Based Development

Liudmyla Koliechkina [1], Olena Dvirna[2] and Serhii Khovben[3]

[1] Algorithms and Databases Department, University of Lodz, Narutowicza 68, Lodz, 90136, Poland
[2] National University "Yuri Kondratyuk Poltava Polytechnic", 24, Pershotravneva Avenue, Poltava, 36011, Ukraine
[3] Poltava University of Economics and Trade, Kovalya street, 3, Poltava, 36000, Ukraine

Abstract

The problems of multi-criteria optimization confidently occupy a prominent place due to their relevance and practical focus. The article is devoted to the formulation of the multi-criteria discrete problem of selecting software components as a multi-criteria combinatorial optimization model based on partial permutations. The method of solving the problem based on the studied combinatorial properties of polyhedra and the graph of the partial permutations set is considered.

Keywords

Combinatorial optimization, combinatorial configurations, multi-criteria model, computer modeling, linear combinatorial optimization

1. Introduction

A natural experiment is an important element in various fields, including design, economy, and management. However, in many cases, a full-scale simulation is not feasible for controlling technological processes in real-time or designing complex systems and devices. Therefore, it is advisable to use computer simulation [1, 2].
Today, computer modeling is utilized in various areas of human activity. Computer modeling is the creating an abstract model process to simulate the behavior and response of a wide systems and prototypes range. The computer modeling software programs quality has increased significantly in the past few years. The computer modeling basis in many cases is a mathematical model use. The main mathematical model purpose in management tasks is to predict the object's response to management influences. In addition, mathematical models are used to study various objects and analyze their sensitivity [3-6].

The mathematical model purposefulness is that it's always built with a specific purpose to solve a practical problem. A computer model is most often understood as a conditional image of an object or some system of objects (or processes), described with the help of interdependent computer tables, schemes, diagrams, graphs, drawings, animation fragments, hypertexts etc., which reflect the structure and relationships between the elements of the object or system.
Mathematical modeling can be considered as a means of studying a real system by replacing it with a more convenient system (model) for experimental research that preserves the essential features of the original. In mathematical modeling, the description function is approximated by a simpler and more convenient function for practical analysis – a model.

Computer modeling is a method of solving an applied problem of analysis or synthesis of a complex system based on the use of its computer model. The essence of computer modeling is to find quantitative and qualitative results with the involvement of an existing mathematical model. A computer model of a complex system should reflect as fully as possible all the main factors and

ProfIT AI 2023: 3rd International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2023), November 20–22, 2023, Waterloo, Canada
lkoliechkina@gmail.com (L. Koliechkina); lenadvirna@gmail.com (O. Dvirna); hovben1996@gmail.com (S. Khovben)
0000-0002-4079-1201 (L. Koliechkina); 0000-0002-0750-6958 (O. Dvirna); 0009-0000-8341-3676 (S. Khovben)

relationships that characterize real situations, criteria and constraints. In addition, the mathematical model should be as universal (to cover the widest possible range of objects close in purpose) as simple (to facilitate the necessary research with minimal costs). The computer direction of modeling in science was called a computational experiment, which is based on the study of a mathematical model with the help of logical-mathematical algorithms and their implementation on a computer.

When solving many practical problems in economics and technology, multi-criteria optimization models are used, where the quality of the solution is evaluated by several criteria at the same time [7-9]. In the simplest interpretation, a multi-criteria problem (MCP) includes objective functions and has no additional constraints. When they are combined into a vector criterion, we arrive at a standard optimization problem. However, an adequate mathematical model of applied problems includes several objective functions, as well as some additional constraints, which make its solution much more complicated than in the simplest version [10-13]. An example of such a model is considered in this paper.

The paper is organized as follows: in the second part, the statement of the applied optimization problem is formulated, the third and fourth parts are devoted to the construction of a mathematical model of the multi-criteria optimization problem on the combinatorial set and the description of the properties of the area of admissible solutions of the problem [14,15]. In the last part, a method for solving such applied problems based on the formulated properties is proposed.

2. Multi-Criteria Decision Discrete Model for Selecting Software Components in Component-Based Development

One of the fundamental modern programming principles is the principle of modularity, which allows for more effective provision of various stages of the software life cycle, such as the creation, implementation and maintenance and improvement of computer software and mathematical support. The modularity principle involves the program development and implementation in the form of a constituent parts set – modules.

The proposed Multi-Criteria Discrete Model (MCDM) for Selecting Software Components in Component-Based Development includes the following steps: identify the software components needed for the project; define the criteria for selecting software components, such as functionality, reliability, compatibility, maintainability, and cost; determine the weights of each criterion based on their importance to the project; evaluate each software component against each criterion using a rating scale; calculate the weighted score for each software component based on the ratings and criterion weights; perform sensitivity analysis to assess the impact of changes in the criterion weights on the rankings. The model considers both the technical and non-technical criteria in software component selection. The weights of the criteria can be determined based on the project requirements and the stakeholders' preferences. The model also allows for flexibility in adjusting the weights to reflect changes in the project's priorities. With this model, software developers can make informed decisions on selecting software components that best meet the project's requirements and constraints.

The MCDM can be formulated as a mathematical model on partial permutations set.

The program can be presented as consisting of separate modules, procedures, programs, segments. For each block, there are possible variants of its implementation in the program. According to the description, the set can be represented as a set of partial permutations.

Let x_i be a binary variable that represents whether software component i is selected or not. $x_i = 1$ if software component i is selected, and $x_i = 0$ otherwise. The decision variables can be defined as follows:

$$x_i \in \{0,\ 1\},\ i\ =\ 1,\ 2,\ \ldots,\ m. \tag{1}$$

If we have n possible components and k possible positions, we will get a set of the partial permutations from m to k A_{nk}^{m} [10].

The objective is to select the best placement of software components. Possible mathematical functions that can be used to quantify the criteria are functionality, reliability, compatibility, maintainability, and cost in the proposed MCDM for Selecting Software Components:

Functionality (f_1) : $f_1 = \langle c_i^1, x_i \rangle \to \max$, where c_i^1 is the level of functionality of software component i, and x_i is the placement of software component i. This function calculates the total functionality score of the selected software components.

Reliability (f_2): $f_2 = \langle c_i^2, x_i \rangle \to \max$, where c_i^2 is the level of reliability of software component i. This function calculates the total reliability score of the selected software components.

Compatibility (f_3): $f_3 = \langle c_i^3, x_i \rangle \to \max$, where c_i^3 is the level of compatibility of software component i. This function calculates the total compatibility score of the selected software components.

Maintainability (f_4): $f_4 = \langle c_i^4, x_i \rangle \to \max$, where c_i^4 is the level of maintainability of software component i. This function calculates the total maintainability score of the selected software components.

Cost (f_5): $f_5 = \langle c_i^5, x_i \rangle \to \min$, where c_i^5 is the cost of software component i. This function calculates the total cost of the selected software components.

These functions can be used to calculate the scores for each of the criteria based on the placement of the software components. The decision maker can then assign weights to each of the criteria and use the weighted sum approach to determine the overall score for each potential solution. Linear constraints can also be added to ensure that the selected software components meet any additional requirements or constraints.

Constraints that limit the total cost of the selected software components can be formulated as follows:

$$\langle a_i^1, x_i \rangle \le b_1 ,$$

where a_i^1 is the cost of software component i, x_i is the placement of software component i, and b_1 is the maximum allowable cost. This constraint ensures that the total cost of the selected software components does not exceed the budget allocated for the project.

Constraints that require a minimum level of functionality or reliability can be formulated as follows:

$$\langle a_i^2, x_i \rangle \ge b_2$$

where a_i^2 is the level of functionality or reliability of software component i, x_i is the placement of software component i, and b_2 is the minimum required level of functionality or reliability. This constraint ensures that the selected software components meet the minimum level of functionality or reliability required for the project.

Various constraints can be added to the overall model as additional linear equations or inequalities. The decision maker can adjust the values of b_1 and b_2 to reflect the specific requirements and constraints of the project.

In addition to these constraints, there may be other linear constraints that can be added to the model depending on the specific requirements of the project. For example, there may be constraints that limit the total cost of the selected software components, or constraints that require a minimum level of functionality or reliability. These constraints can be formulated as linear equations or inequalities and incorporated into the overall model.

A linear constraint can be added to the proposed MCDM for Selecting Software Components to incorporate security requirements. One way to do this is to assign a security score to each software component based on its level of security, and then add a linear constraint that ensures that the total security score of the selected components meets a minimum threshold.

The security score for software component i can be denoted as a_i^2, and the minimum required security score can be denoted as b_3. The linear constraint can be formulated as:

$$\left\langle a_i^3, x_i \right\rangle \geq b_3,$$

where x_i is the placement of software component i [10].

This constraint ensures that the total security score of the selected software components is at least S, indicating that the selected components meet the minimum security requirements for the project. The decision maker can adjust the value of S to reflect the specific security needs of the project. In addition to this constraint, there may be other linear constraints that can be added to the model to further incorporate security requirements. For example, there may be constraints that limit the maximum allowable security risk or that require specific security features. These constraints can be formulated as linear equations or inequalities and incorporated into the overall model.

The mathematical model of described task based on the proposed Multi-Criteria Discrete Model for Selecting Software Components: to optimize objective functions of $x_i \in E_{nk}$

$$\begin{aligned} f_1 &= \left\langle c_i^1, x_i \right\rangle \to \max, \\ f_2 &= \left\langle c_i^2, x_i \right\rangle \to \max, \\ f_3 &= \left\langle c_i^3, x_i \right\rangle \to \max, \\ f_4 &= \left\langle c_i^4, x_i \right\rangle \to \max, \\ f_5 &= \left\langle c_i^5, x_i \right\rangle \to \min \end{aligned} \tag{2}$$

subject to:

$$\left\langle a_i^1, x_i \right\rangle \leq b_1,\ \left\langle a_i^2, x_i \right\rangle \geq b_2,\ \left\langle a_i^3, x_i \right\rangle \geq b_3. \tag{3}$$

The objective function is a weighted sum of the scores for each criterion, where the weights reflect the relative importance of each criterion. The decision maker can adjust the weights to reflect the specific priorities and preferences of the project. The constraints ensure that the selected software components meet the minimum requirements for functionality, reliability, compatibility, maintainability, and security, and that the total cost of the selected components does not exceed the allocated budget. The decision variables xi are binary variables indicating whether each software component is selected or not selected.

The proposed model provides a quantitative approach to software component selection in component-based development. The decision maker can make informed decisions based on the mathematical model and adjust the weights of the criteria to reflect changes in the project's requirements and constraints.

This problem is a model of a multiobjective optimization problem with additional constraints on a combinatorial set of partial permutations and can be solved by a combined approach.

3. Mathematical model of the applied problem and its interpretation in terms of multiobjective combinatorial optimization

Consider our model as multiobjective optimization problem on a set of partial permutations and investigate its properties. To do this, we will use the concept of mapping combinatorial sets into Euclidean space [16-18], The problem is to optimize several criteria $F=\{f_1(x), f_2(x),..., f_L(x)\}$ on a finite set X , which can be represented as:

$$f_l(x) \to \min,\ l \in J_{L'} = \{1,...,L'\};$$

$$f_l(x) \to \min,\ l \in J_L \setminus J_{L'}; \tag{4}$$

$$x \in X \subseteq E', \tag{5}$$

where E' is a combinatorial space, X is a set of possible solutions and functions $f_l(x)$, $l \in J_L$, which are defined on E'.

The main capabilities of multiobjective optimization are aimed at solving the problem of effective search for a solution [7-9, 11]. The first attempt to formulate the concept of efficient solutions was made by V. Pareto [7-9], and its set of such solutions is called the Pareto set. However, it is not always possible to find all effective sets of solutions when solving problems and applying multiobjective methods. Therefore, it is sometimes important to define at least part of the Pareto set, as well as its extension. At the same time, we can talk about Smale's or Slater's set, etc. [11].

By solution $Z(F,X)$ we mean an element or elements of one of the following sets [6-9]:

1. The set $I(F,X)$ of ideal solutions:

$$I(F,X) = \{x \in X : v(x,F,X) = \varnothing\}, \tag{6}$$

where $v(x,F,X) = \{y \in X \mid \exists l \in J_L : f_l(y) > l_i(x)\}$;

2. Pareto set $P(F,X)$, that is, a set of (Pareto optimal) solutions:

$$P(F,X) = \{x \in X : \pi(x,F,X) = \varnothing\}, \tag{7}$$

where $\pi(x,F,X) = \{y \in X : F(y) \geq F(x), F(y) \neq F(x)\}$;

3. Slater's set $Sl(F,X)$ of inefficient solutions:

$$Sl(F,X) = \{x \in X : \sigma(x,F,X) = \varnothing\}, \tag{8}$$

where $\sigma(x,F,X) = \{y \in X : F(y) > F(x)\}$;

4. Smailer's set $Sm(F,X)$ of strictly efficient solutions:

$$Sm(F,X) = \{x \in X : \eta(x,F,X) = \varnothing\}, \tag{9}$$

where $\eta(x,F,X) = \{y \in X \setminus \{x\} : F(y) \geq F(x)\}$.

For example, an element of set (6) is called an ideal solution [11], and it is the best according to all specific criteria, respectively, and for multiobjective problem. At the same time, Pareto optimality (see (7)) means that the value of any of the specific criteria can be increased only by decreasing the value of at least one of the other specific criteria. For a weakly efficient estimate/decision (8), there will be no such estimate/decision that would be better by all specific criteria. As a result, the sets (6)-(9) are related as follows:

$$I(F,X) \subset Sm(F,X) \subset P(F,X) \subset Sl(F,X)\}. \tag{10}$$

It should be noted that, as a rule, Pareto-optimal solutions (efficient solutions) are searched for when solving the PMO. Today, there are several directions of development of multi-criteria optimization methods, which, conditionally, can be divided as follows: methods based on the criteria of collapsing into a single criterion (actual reduction to a single-criteria problem); methods based on imposing restrictions on criteria; methods based on finding a compromise solution; methods that use human-machine decision-making procedures or interactive programming.

Most methods of constructing a set of effective solutions use certain optimality conditions. Necessary conditions are often applied, for example, if the point is effective. Thus, the most common methods of the PMO are the method of reducing a multi-criteria problem to a single-criteria one by collapsing a vector criterion into a super-criteria, the method of priorities, and their generalization — the method of successive concessions [7-9,11]. With the help of the first method, we can reduce a multi-criteria problem to a single-criteria problem, the other two make it possible to reduce the original problem to a sequence of single-criteria optimization problems.

So, let's consider the reduction of a multiobjective problem to a linear optimization problem, using the supercriterion as a weighted sum of $F(X)$ - components:

$$\Phi(x)=\sum_{l=1}^{L}\alpha_l f_l(x)\ ,\ \alpha_l \le 0\ ,\ l\in J_{L'}\ ,\ \alpha_l \ge 0\ ,\ l\in J_L \setminus J_{L'}\ ,\ \sum_{l=1}^{L}|\alpha_l|=1. \tag{11}$$

So, the problem is considered further:

$$\Phi(x)\to\max;\ x\in X\subseteq E'\subset R^N. \tag{12}$$

When performing convolution (11), the main issue is the correct choice of coefficients α_l, $l\in J_L$, the relative importance of decision-making criteria x^* from (5) coincide with the solution x' from (12) [9, 11].

Individual criteria are ordered according to their relative importance – $f_{i_1}\succ f_{i_2}\succ\ldots\succ f_{i_L}$.

Then the first, most important criterion is maximization and limitation

$$f_{i_1}(x)\ge f_{i_1}^*-\Delta_1=f_{i_1}'$$

on the lower limit $f_{i_1}(x)$ is added, where $f_{i_1}^*=\min_{x\in X} f_{i_1}(x)$, $\Delta_1\ge 0$ – is a concession on $f_{i_1}^*$.

Next, the second most important criterion is optimized on the new domain

$$X^1=\{x\in X: f_{i_1}(x)\ge f_{i_1}'\}, \text{ etc.}$$

As a result, a number of problems:

$$f_{i_1}(x)\to\min,\ x\in X^{l-1}\subseteq E',\ l\in J_L, \tag{13}$$

where $X^0=X$,

$$X^l=\{x\in X^{l-1}: f_{i_{1'}}(x)\ge f_{i_l}', l'\in J_{l-1}\}, \tag{14}$$

$$f_{i_l}'=f_{i_l}^*-\Delta_l,\ f_{i_l}^*=\min_{x\in X^{l-1}} f_{i_l}(x), \Delta_l\ge 0,\ l\in J_{L-1}.$$

x^* comes out as a solution to the last of them.

As mentioned above, the model of the multiobjective optimization problem is considered on the combinatorial set, i.e. it also belongs to the class of combinatorial optimization problems.

Practice has shown that most applied problems are set on a set of Boolean vectors or permutations [10]. This means that E' is usually an underlying Boolean value $C-set$ (C_b-set) B_n from n-dimensions Boolean vectors or main C_b-set permutations $E_{nk}(G)$ included n-elements of the numerical multiset $a=\{a_1,\ldots,a_n\}$, $g_1\le\ldots\le g_n$, containing k various elements [10, 16].

A specific case on $E_{nk}^m=E_{nk}^m(A)$ is considered at the moment at $k=n$, where A – this is a set, $E_{nk}^m=E_{nk}^m(A)$ - partial permutations with repetitions.

It $E_n(A)$ called C_b-set permutation without repetitions or simply C_b-set permutation.

An interesting feature of this set is that E', and accordingly X, coincides with the set of vertices of its convex hull:

$$E'=vert\,P',\ P'=conv\ E'.$$

The set X can be sets of Euclidean combinatorial configurations of partial permutations, permutations, polypermutations, and others.

The properties of Euclidean combinatorial configurations are the basis for the development of methods for solving the given problems. One of the important areas is the connection of combinatorial configurations with combinatorial polyhedra and their graphs. This representation of Euclidean combinatorial configurations allows structuring sets e - configurations to analyze the values of the functions on them [10,19,20]. Let's consider this connection in more detail.

4. Properties of graphs of Euclidean combinatorial configurations sets

The set of such Euclidean combinatorial configurations as configurations of permutations, partial permutations, polypermutations, polyplacements coincide with the set of vertices of the corresponding polyhedra [21,22].

For example, the convex hull of the general set of e -configurations of partial permutations $conv\, E_n^m(A)$ is a polyhedron of partial permutations $M_n^m(a)$, which is described by systems :

$$\begin{cases} \sum\limits_{i \in w} x_i \geq \sum\limits_{j=1}^{|w|} a_j \quad \forall w \subset J_k \ |w| < k \\ \sum\limits_{i=1}^{k} x_i = \sum\limits_{j=1}^{k} a_j , \end{cases}$$

and symmetrical to it:

$$\sum_{i \in w} x_i \leq \sum_{j=1}^{|w|} a_{k-j+1} \quad \forall w \subset J_k ; \ |w| < k \ ,$$

$$\sum_{i=1}^{k} x_i = \sum_{j=1}^{k} a_j \, .$$

Polyhedron of partial permutations $E_n^m = E_n^m(A)$, $a = \{a_1,...,a_n\}$, is the set of all solutions of the system of linear inequalities described by relations [11,23,24].

The properties of the polyhedron $M_n^m(a)$ are used to construct methods for solving optimization problems on combinatorial configurations. Vertices and edges of a polyhedron can be represented in the form of a corresponding graph.

Important results are given by the connection of combinatorial configurations [25] with the theory of graphs, which was studied in [11, 14, 26].

Many discrete optimization problems are presented in terms of graph theory. Theoretical-graphic models are most widely used in the field of construction and research of communication networks, in the study of chemical and genetic structures, electrical circuits, etc.

A graph G is a figure that can be represented by a pair G, where $V = \{v_1, v_2, ..., v_m\}$ – is not an empty finite set of vertices, and $U = \{u_1, u_2, .., u_n\}$ – a finite set of directed or undirected edges connecting pairs of vertices. Let the edge connect the vertices v_i, v_j, i.e $u_{ij} = (v_i, v_j)$, then they are adjacent vertices incident to one edge u_{ij}.

Let X – he set of e-configurations, and $G(V,U)$ – some graph for which the number of vertices coincides with the cardinality of the set X. Let's carry out an objective mapping of ψ

elements of the set $X \subset R^n$ into the set V vertices of the graph G, that is, to each element $x \in X \subset R^n$ let's match $v \in V$, thus we have $G^X(V,U)$ – graph of the set of e-configurations X.

We will consider the designations $G^X(V,U)$ and G^X to be identical.

Definition 1. If there is a bijective mapping $\psi : X \to V$, where X – set of e-configuration, and V – set of vertices of some graph G^X, and also defined Ψ – conditions of adjacency of vertices, that G^X is a graph of the set of e-configurations X.

For convenience, we will consider the vertices of the graph G^X as corresponding elements of e-configurations, that is, the vertices are mapped into the Euclidean space [3, 10].

For a set of partial permutations, the complexity arises when the set ceases to be vertices, that is, some points of e-configuration are contained on the edges of the polyhedron, which is also displayed in the graph. The question arises of constructing such a graph or a set of graphs that would allow considering all e- configurations as vertices.

If we define a different condition for the adjacency of the vertices of the e-configuration graph than the one given above, we will obtain a new graph.

Definition 2. Vertices of the graph G^X, adjacent to the vertex we call those and only those vertices obtained from x by permuting arbitrary components of the inducing set $e_i, e_j \forall i, j \in J : i \neq j$.

Definition 3. A transposition graph G^X_{Tr} let's call such a graph G^X, whose adjacent vertices are defined according to Definition 2.

Based on the transposition graph, we will construct a grid graph of the Euclidean combinatorial configuration, having previously entered the corresponding definitions.

Construction and properties of the structural graph of e-configurations.

The work [10] used the concept of a structural graph of combinatorial configurations, the vertices of which are only some points of the set of Euclidean combinatorial configurations that satisfy the set requirements. Let's formulate the definition and consider its properties.

Let the subsets X' sets of combinatorial configurations X are determined by h fixed coordinates, and the elements of the set have the form $x = (x_1, x_2, ..., x_m)$.

Definition 4. Let $G^{X'}(\tilde{V},\tilde{U})$ – subset graph X' combinatorial configurations X, and the plural X' structured in such a way that $X' = X_1' \cup X_2' \cup ... \cup X_\lambda'$ and each subset $X_i', i \in J_\lambda$ corresponds to vertices h with fixed coordinates and is represented by two vertices x_i^0 та x_i^{st} such that the conditions for a linear function $f(x)$ are fulfilled:

$$f(x_i^0) = \max_{x \in X_i'} f(x),$$

$$f(x_i^{st}) = \min_{x \in X_i'} f(x), \tag{16}$$

and the edges of the graph $G^{X'}(\tilde{V},\tilde{U})$ are those that connect the corresponding vertices x_i^0 and x_i^{st} and the vertices are formed by successive transpositions of fixed coordinates, then we will call such a graph the structural graph of the set of Euclidean combinatorial configurations and denote $n\ G^X_S(\tilde{V},\tilde{U},h)$.

Definition 5. Vertices x_i^0 and x_i^{st} subsets $X_i', i \in J_\lambda$ of the structural graph $G_S^X\left(\tilde{V},\tilde{U},h\right)$, for which the conditions (16) are fulfilled are called the peaks of leakage and drain, respectively $G_S^X\left(\tilde{V},\tilde{U},h\right)$, and the quantity λ will be called the level of the structural graph.

Let's consider several examples of a structural graph illustrating the main stages of their construction.

Example 1. Let X – be the set of Euclidean combinatorial configurations of permutations from the set $A=\{1,2,3,4\}$. Let's put $h=1$, that is, we fix only one coordinate, then we get four subsets such that $x_4^1=4, x_4^2=3$, $x_4^3=2$, $x_4^4=1$.

We will get four pairs of vertices, respectively x_i^0 and x_i^{st}: $x_1^0=(3,2,1,4)$, $x_1^{st}=(1,2,3,4)$, $x_2^0=(4,2,1,3)$, $x_2^{st}=(1,2,4,3)$, $x_3^0=(4,3,1,2)$, $x_3^{st}=(1,3,4,2)$, $x_4^0=(4,3,2,1)$, $x_4^{st}=(2,3,4,1)$. Let's mark the vertices on the graph $G_S^X\left(\tilde{V},\tilde{U},1\right)$, shown in the figure 2.3. We also denote the subgraphs into which the structural graph is divided.

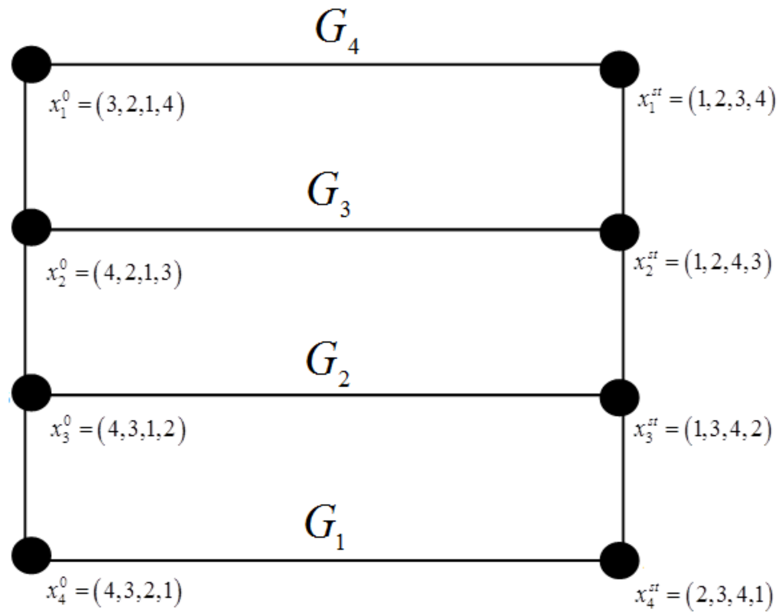

Figure 1: Structural graph $G_S^X\left(\tilde{V},\tilde{U},1\right)$ sets of Euclidean combinatorial configurations of permutations

Figure 1 shows the levels of the structural graph G_4,G_3,G_2,G_1, whose numbers correspond to fixed coordinates.

Example 2. Let us have X – a set of Euclidean combinatorial configurations of partial permutations of dimension 4 from the set $A=\{1,2,3,4,5,6\}$.

Let's put $h=2$, that is, we will fix the last two coordinates. Let's get the following set of pairs of coordinates:

$$\{(6,5),(6,4),(6,3),(6,2),(6,1),(5,6),(5,4),(5,3),(5,2),(5,1),(4,6),(4,5),(4,3),(4,2),(4,1),$$
$$(3,6),(3,5),(3,4),(3,2),(3,1),(2,6),(2,5),(2,4),(2,3),(2,1),(1,6),(1,5),(1,4),(1,3),(1,2)\}$$

So, we get thirty levels of the structural graph and the corresponding number of pairs of vertices x_i^0 and x_i^{st}. Let's mark them selectively in the Figure 2.

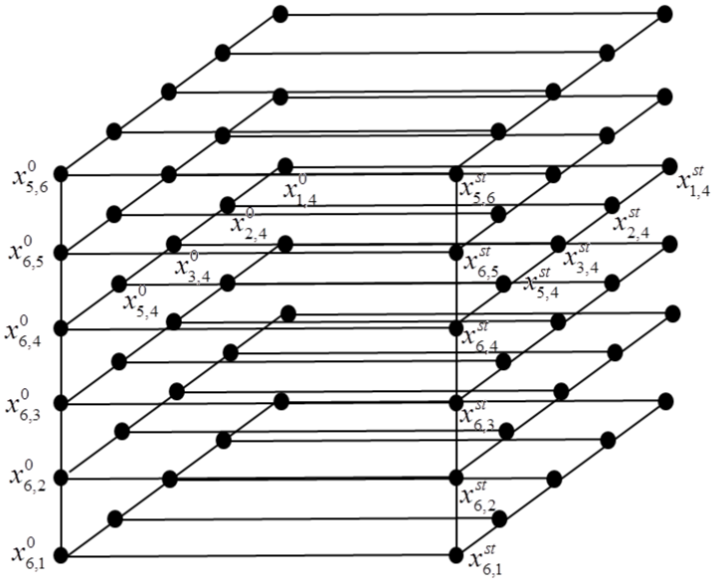

Figure 2: Structural graph $G_S^X\left(\tilde{V},\tilde{U},2\right)$

The use of grid graphs and structural graphs makes it possible to build new methods of solving the vector problem of linear combinatorial optimization, which are models of applied problems [10, 11, 14]. Thus, using the properties of combinatorial sets graphs, it is possible to reduce the number of necessary steps for solving problem (4)-(5) and quickly find the solution itself, which will be some subset of effective solutions.

For problem (4)-(5), one can use methods aimed at finding one effective solution, which is quite sufficient for most practical problems.

5. Method of the solving multiobjective optimization problem on a combinatorial set of partial permutations

Let us consider the approach to solving optimization problems on combinatorial configurations of partial permutations. Given that the problem is multi-criteria, it is advisable to use vector optimization methods.

Thus, at the first step, the method of successive concessions is used for problems with many criteria. The description of the next stage directly depends on the specifics of the combinatorial optimization problem. In this case, it is quite productive to use methods based on the properties of an approximate graph, in particular, the horizontal method. So, when using the method of successive concessions and the horizontal method, we will get a new approach that describes the main properties of combinatorial configurations and features of multi-criteria optimization. We will describe it using the following steps.

1. Let's enter the input parameters: elements of the generating set A, optimality criteria

$$f_i\ (x) \to \min, i \in J_m,$$

linear constraints $g_i\,(x) \geq b_i\,, i \in J_k$.

2. For each constraint, we will construct a structural directed graph and obtain sets $D_i, i \in J_k$.

3. Let's find combinations of sets of partial permutations according to the formula $A_n^k = \dfrac{n!}{(n-k)!} = \dfrac{P_n}{P_{n-k}}$. As a result of calculations, we will get a graph containing A_n^k vertices.

4. Let's calculate the coefficient Δ, which is determined by the formula:

$$\Delta = (g_1 a_1 + g_2 a_2 + \ldots + g_n a_k) - b_i,$$

where g_n – coefficient of the set of partial permutations; a_k – coefficient of the limiting function; b_i – border to limit a_k.

It should be noted that limiting functions are usually used in extreme problems, which serve as additional parameters for sampling admissible values [2]. In such cases, it is advisable to introduce indicators that are designed to select only those subgraphs that can contain a potential answer. The effectiveness of the described approach lies in the method of permutations, which significantly reduces the calculation time. It makes it possible to discard subgraphs that do not satisfy the conditions of the algorithm. Thanks to this concept, calculations are performed only on subgraphs for which the conditions are fulfilled:

$$A_x \geq b, \quad A_x - b \geq 0.$$

Thus, the algorithm for solving extreme problems on combinatorial configurations of partial permutations is reduced to gradually deepening the graph and dividing it into subgraphs. At the same time, thanks to the introduction of additional parameters, those subgraphs that do not satisfy the restriction conditions are discarded. It is expedient and relevant to continue the research of combinatorial problems on multiple partial permutations in order to implement algorithms that will ensure maximum performance of calculations by introducing additional parameters.

6. Conclusions

The article presents a multi-criteria decision discrete model for selecting software components in component-based development and proposes an approach for solving such a problem.

The solution to the problem lies in the preliminary study of the extremal properties of combinatorial configurations and their graphs, which are the area of search for solutions with further application in the combined method.

References

[1] Kaltdorf, M., Breitenbach, T., Karl, S. et al. Software JimenaE allows efficient dynamic simulations of Boolean networks, centrality and system state analysis. Sci Rep 13, 1855 (2023). https://doi.org/10.1038/s41598-022-27098-7

[2] Emmerich, M., Deutz, A., 2018. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. Nat. Comput. 17.

[3] Stoyan, Y.G., Yakovlev, S.V. Configuration Space of Geometric Objects. Cybern Syst Anal 54, 716–726 (2018). https://doi.org/10.1007/s10559-018-0073-5

[4] Mashtalir, V.P., Shlyakhov, V.V., Yakovlev, S.V. Group Structures on Quotient Sets in Classification Problems. Cybern Syst Anal 50, 507–518 (2014). https://doi.org/10.1007/s10559-014-9639-z

[5] Gerasin, S.N., Shlyakhov, V.V., Yakovlev, S.V. Set coverings and tolerance relations. Cybern Syst Anal 44, 333–340 (2008). https://doi.org/10.1007/s10559-008-9007-y

[6] Koliechkina and A. Nahirna, "Mathematical Model on Set of Partial Permutations for the optimization of Complex Information Systems," 2020, IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC), Kyiv, Ukraine, 2020, pp. 1-4, doi: 10.1109/SAIC51296.2020.9239175.

[7] Bökler, F. K., 2018. Output-Sensitive Complexity of Multiobjective Combinatorial Optimization with an Application to the Multiobjective Shortest Path Problem (Ph.D. thesis). Technische Universität Dortmund.

[8] Odu, G. O., Charles-Owaba, O. E. (2013). Review of Multi-criteria Optimization Methods – Theory and Applications. IOSR Journal of Engineering (IOSRJEN), vol. 3, Issue 10, pp. 1 – 14.

[9] Panos M. Pardalos, Antanas Žilinskas, and Julius Žilinskas. Non-Convex Multi-Objective Optimization. Springer Optimization and Its Applications. Springer International Publishing, 2017.
[10] Pichugina O., Koliechkina L. (2021) Linear constrained combinatorial optimization on well-described sets. IOP Conf. Series: Materials Science and Engineering 1099 (2021) 012064 IOP Publishing doi:10.1088/1757-899X/1099/1/012064
[11] Koliechkina, L.N., Dvirna, O.A. & Khovben, S.V. A Two-Step Method for Solving Vector Optimization Problems on Permutation Configuration. Cybern Syst Anal 57, 442–454 (2021). https://doi.org/10.1007/s10559-021-00369-3
[12] Korte B, Vygen. J. Combinatorial Optimization: Theory and Algorithms. Springer, 5th edition, 2012.
[13] Pardalos P.M. Handbook of combinatorial optimization / P.M. Pardalos, D-Z. Du, R.L.Graham. – New York: Springer, 2013. – 3409 p.
[14] Donets, G.P., Koliechkina, L.N., Nahirna, A.N. A Method to Solve Conditional Optimization Problems with Quadratic Objective Functions on the Set of Permutations. Cybernetics and Systems Analysis. 2020. Vol. 56, N 2. P. 278-288.
[15] B. Korte, J. Vygen, Combinatorial Problems in Chip Design, in Building Bridges, Springer, Berlin, Heidelberg, 2008, Pp. 333–368.
[16] Stoyan, Y.G., Yakovlev, S.V. Theory and Methods of Euclidian Combinatorial Optimization: Current Status and Prospects. Cybern Syst Anal 56, 366–379 (2020). https://doi.org/10.1007/s10559-020-00253-6
[17] Yakovlev, S. Convex Extensions in Combinatorial Optimization and Their Applications. Optimization Methods and Applications . Springer Optimization and Its Applications, 2017, vol 130. Springer, Cham. https://doi.org/10.1007/978-3-319-68640-0_27
[18] Pichugina, O.S., Yakovlev, S.V. Continuous Representations and Functional Extensions in Combinatorial Optimization. Cybern Syst Anal 52, 921–930 (2016). https://doi.org/10.1007/s10559-016-9894-2
[19] Yakovlev, S.V. Bounds on the minimum of convex functions on Euclidean combinatorial sets. Cybern Syst Anal 25, 385–391 (1989). https://doi.org/10.1007/BF01069996
[20] Stoyan, Y.G., Yakovlev, S.V., Emets, O.A. et al. Construction of convex continuations for functions defined on a hypersphere. Cybern Syst Anal 34, 176–184 (1998). https://doi.org/10.1007/BF02742066
[21] Yakovlev, S.V., Pichugina, O.S. Properties of Combinatorial Optimization Problems Over Polyhedral-Spherical Sets. Cybern Syst Anal 54, 99–109 (2018). https://doi.org/10.1007/s10559-018-0011-6
[22] Pichugina O. and Yakovlev, S. Optimization on polyhedral-spherical sets: Theory and applications, 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, Ukraine, 2017, pp. 1167-1174, doi: 10.1109/UKRCON.2017.8100436.
[23] Yemelichev V.A., Kovalev M.M., Kravtsov M.K. Polytopes, graphs and optimisation. Cambridge University Press, Cambridge. 1984. 243p.
[24] Pichugina, O., Yakovlev, S. Functional and analytic representations of the general permutations. Eastern-European Journal of Enterprise Technologies, 2016, 1(4), pp. 27–38, http://dx.doi.org/10.15587/1729-4061.2016.58550
[25] Pichugina O., Yakovlev, S. Euclidean Combinatorial Configurations: Typology and Applications, 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), Lviv, Ukraine, 2019, pp. 1065-1070, doi: 10.1109/UKRCON.2019.8879912.
[26] Koliechkina L.N., Nagornaya A.N., Semenov V.V. Method of solving problem of conditional optimization on combinatorial set of arrangements. Journal of Automation and Information Sciences. 2019. 51, N.8. P. 31-42.