Multiobjective Optimization on Permutations with Applications

Liudmyla KOLIECHKINA\(^1,\)\(^*\) and Oksana PICHUGINA\(^2,\)\(^*\)

\(^1\) University of Lodz, Uniwersytetka Str. 3, 90-137 Lodz, Poland
\(^2\) National Aerospace University Kharkiv Aviation Institute, Chkalova Str. 17, 61070 Kharkiv, Ukraine
\(^*\) Corresponding author

Keywords: Multiobjective Combinatorial Optimization, Vector Optimization, permutation, structural graph, combinatorial configuration.

Abstract. A method of multiobjective optimization on permutations (MOP) is offered based on the Directed Structuring Method and using Graph Theory. Prospects of applying graph techniques are caused by representability of the feasible domain by graph vertices. It yields advantages in using traditional methods, as well as in developing new ones. Our method is a generalization of the Method for Sequential Analysis of Variants for multiobjective optimization on permutations and multi-permutations. Most problems on combinatorial configurations sets are NP-hard, and a search of an exact solution requires enumerating a factorial number of variants. To decrease it, the method includes: a choice of an unconstraint MOP method; a choice of a method for generating a sequence of feasible solutions for a constraint MOP adapted to objective function; constructing and examining a structural graph of the optimization problem; a polynomial algorithm choice for solving the problem on partially ordered vertices of the graph.

Introduction

From the practical point of view, a wide and important class of decision-making problems is multiobjective problem, where the quality of a decision is evaluated regarding several criteria simultaneously [11, 12, 19].

Problems of optimization of several functions arise in the study of many theoretical and practical problems. Any problem of optimal design of complex economic and technical systems, schemes, technological devices, structures, scheduling, planning, and management of production activity, etc. requires the solution found takes into account many criteria and constraints [4–6,10,23]. This problem is a multiobjective optimization one (MO problem, MP).

Research in the field of multiobjective optimization is currently intensively stimulated by practical needs and development of computer information technology. This is the origin of a large number of papers devoted to MO [4,6,7,10,22,25].

In the simplest interpretation, MP includes objective functions and no additional constraints. When they are combined into a vector criterion, we come to a standard optimization problem. However, an adequate mathematical model of real-world problems includes several objective functions, as well as some additional constraints making it solving much more complicated than the simplest one.

In [4,6,7,10,22,25], methods for solving multiobjective optimization problems are considered, both constrained and unconstrained. In such studies [4,6,7,10,22,25], various approaches to
their solution are offered. In particular, methods of a search and examining the whole feasible domain are described in [4,10,20].

Since real-world systems are discrete-continuous by nature, they are modeled as partially integer programs. In particular, numerous problems of planning, management, design, and placement are modeled with the help of multiobjective problems whose solutions are of combinatorial nature, e.g., permutations, partial permutations, combinations, compositions, partitions, as well as their composition images [5,7,8,22,24]. In this case, the search for an optimal solution is conducted on the correspondent combinatorial set or its proper subset. MPs in their own are complex, but their solving becomes much more complicated if solutions are sought on a combinatorial set, and a multiobjective combinatorial optimization problem (a multiobjective combinatorial problem, a multiobjective CP, MCP) needs to be solved. MCP approaches take mainly are based on structural peculiarities of a particular combinatorial set. In [7,8], a multiobjective optimization on permutations without repetitions is considered. The formulation of MCP on poly-partial permutations is formulated in [22,23], and an approach to its solution using polyhedral relaxation is offered from the group of cutting methods. In [24], the MCP formulation of linear-fractional optimization is presented, and a solution method is offered, which is based on explored properties of a feasible combinatorial domain. A connection between the linear combinatorial optimization problem (linear CP) and linear-fractional MCP is established for the case if the feasible domain is a set of combinations. It is well-known that standard combinatorial optimization methods typically are not applicable to problems with many criteria [22]. An advantage of the approach is that it allows applying classical optimization methods to solving MCP.

In studies [1–3], methods for solving CP are presented, which are based on the properties of graphs of polytopes of combinatorial sets. Such methods have advantages over classical ones. Since using the properties of the polytopes, the number of iterations (search operations) decreases, and the search itself is carried out along the vertices of the directed graph of the polytope, which vertex set represents the corresponding combinatorial set [2].

In [24], the MCP problem with a linear-fractional objective function on the set of permutations is solved by the coordinate method of localizing the value of a linear function for the solution of CP (the coordinate localization method, CLM). CLM is based on the construction of the CP graph and its decompositions into subgraphs and viewing a limited number of them.

Note that CLM was originally developed to a feasibility problem with an equality constraint. Adapted to a COP, CLM allows reducing significantly the number of combinatorial configurations considered [1,3].

The goal of this paper is to develop a method for solving linear CP (LCP) on the general set of permutations [31] based on adapting CLM to the problem and to develop an approach to linear multiobjective optimization on the combinatorial set. For the case under consideration, CLM requires a modification (further referred to as a modified CLM (MCLM)) and development.

The extension from standard permutations without repetitions to multi-permutations, where a multiplicity of their components is permitted, allows covering a much wider class of MOPs, including boolean permutations appearing in modeling numerous graph problems [15]. Also, the method presented can be considered as a new approach to LCP on standard permutations. Also, the extension cover much wider class if LCPs than CLM.

The paper is organized as follows. In the second section, a multiobjective problem of Euclidean Combinatorial Optimization is formulated, and the conditions on the existence of a set of solutions are listed. Section 3 is dedicated to the analysis of methods and approaches to solving multicriteria
The Multiobjective Problem of Euclidean Combinatorial Optimization

MCP consists in optimizing several criteria \( \{f_1(x), f_2(x), \ldots, f_L(x)\} \) on a finite set \( X \), i.e., it can be represented as:

\[
(MCP) \quad f_l(x) \rightarrow \min, \ l \in J_L = \{1, \ldots, L\}; \\
(MCP) \quad f_l(x) \rightarrow \max, \ l \in J_L \setminus J_L'; \\
x \in X \subseteq E',
\]

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

It is convenient to represent the criteria in the form of a vector function (a vector criterion) \( F(x) = \{f_k(x), \ k \in J_L\} \). As a result, (1) takes a form:

\[
F(x) \rightarrow \text{extr}, x \in X \subseteq E'.
\]

Without loss of generality, we can assume that (2) has the form

\[
(Z(F,X)) : F(x) \rightarrow \max, x \in X \subseteq E',
\]

and \( F = (-f_1(x), \ldots, -f_{L'}(x), f_{L'+1}(x), \ldots, f_L(x)) \).

Note that each solution \( x = (x_1, x_2, \ldots, x_n) \in X \) is characterized by a relevant vector estimate, that is, a vector \( F(x) \). Therefore, a choice of the optimal solution is, in fact, a choice of an optimal estimate from the set of estimates:

\[
Y = F(X) = \{y \in \mathbb{R}^L \mid y = F(x), x \in X \}.
\]

In this case, the effectiveness of the estimates (and solutions) is determined by a chosen principle of optimality.

Suppose that the combinatorial space \( E' \) is a non-empty finite set of points \( \mathbb{R}^n \), and MCP \( Z(F,X) \) with \( F = (f_1, \ldots, f_L) \) is considered. If a set \( E' \) is an image of a certain set \( A \) of real combinatorial objects in Euclidean space provided that between elements \( E' \) and \( A \) a bijection is established, then the set \( A \) is Euclidean combinatorial set (e-set), and \( E' \) is the corresponding set of Euclidean combinatorial configurations (C-set) [26]. Respectively, (3) belongs to the class of multiobjective problems of Euclidean Combinatorial Optimization (multiobjective ECOP, MECOP) [23][26]. Moreover, it is a general mathematical model of MECOP.

Let us consider \( Z(F,X) \). Assume that a feasible domain \( X \) of the problem is non-empty and is can be represented as follows \( X = \{x \in E \subseteq \mathbb{R}^n : x \geq 0, G(x) \leq b \} \neq \emptyset \). This implies that there exists a solution of the problem \( Z(F,G) \) for each component

\[
f_l(x), l \in J_L,
\]

of a vector criterion \( F(X) \). Also, let us assume that the multiobjective problem \( Z(F,G) \) is such that extreme points \( x^k \) of particular problems of optimizing \( [5] \) are not the same (otherwise, an
ideal solution of the problem exists). From this, it follows that a solution of the vector problem \( Z(F,G) \) is a compromise satisfying in some way all components of the vector criterion. Therefore, solutions to the problems are not optimal, but effective.

To solve the problem of effective solution search, main affords in multiobjective optimization are directed \([4–6, 11, 12, 19]\). The first attempt to formulate a concept of effective solutions was done by V. Pareto (see, for instance, \([12, 19]\)), and his set of such solutions is called Pareto set. However, when applying multiobjective methods, other effective solution sets can be more suitable. They can be restrictions of the Pareto set, as well as its extension. When applying one method of multicriteria choice, Pareto set can be narrowed or expanded becoming Smale set, Slater set, etc. \([12]\).

Under a solution of \( Z(F,G) \) we mean an element or elements of one of the following sets \([22–24]\):

1. The set \( I(F,X) \) of ideal solutions:
   \[
   I(F,X) = \{ x \in X : \nu(x,F,X) = \emptyset \},
   \]
   where \( \nu(x,F,X) = \{ y \in X | \exists l \in J_L : f_l(y) > l_i(x) \} \); \(6\)

2. Pareto set \( P(F,X) \), that is, sets of effective (optimal Pareto) solutions:
   \[
   P(F,X) = \{ x \in X : \pi(x,F,X) = \emptyset \},
   \]
   where \( \pi(x,F,X) = \{ y \in X : F(y) \geq F(x), F(y) \neq F(x) \} \); \(7\)

3. Slater set \( Sl(F,X) \) of weakly effective solutions:
   \[
   Sl(F,X) = \{ x \in X : \sigma(x,F,X) = \emptyset \},
   \]
   where \( \sigma(x,F,X) = \{ y \in X : F(y) > F(x) \} \}; \(8\)

4. Smale set \( Sm(F,X) \) of strictly effective solutions:
   \[
   Sm(F,X) = \{ x \in X : \eta(x,F,X) = \emptyset \},
   \]
   where \( \eta(x,F,X) = \{ y \in X \setminus \{ x \} : F(y) \geq F(x) \} \}. \(9\)

For instance, an element of the set \([6]\) is called an ideal solution \([4,10,23,25]\), and it is the best for all the particular criteria, respectively, for MP as well. At the same time, Pareto optimality (see \([7]\)) means that the value of any of the particular criteria can be increased only by reducing the value of at least one of the other particular criteria. For a weakly effective estimate/solution \([8]\) there will be no such estimate/solution that would be better with respect to all particular criteria. As a result, the sets \([6–9]\) are connected as follows:

\[
I(F,X) \subset Sm(F,X) \subset P(F,X) \subset Sl(F,X).
\]

It should be noted that, as a rule, when solving MPs, Pareto optimal solutions (effective solutions) are sought. We are not an exception.

**Approaches to Multiobjective Euclidean Combinatorial Optimization**

The problem of finding all effective solutions is not only of theoretical but also a great practical interest. It is explained by the fact that the construction of the whole Pareto set or its rather
broad subsets is one of the first stages in the whole series of optimal selection procedures under several criteria \cite{10,23,24}. To solve the problem of constructing the set, one can choose two ways - to search directly on a feasible domain or to introduce a parametrizing set first and then search within it.

There are several directions of developing multiobjective optimization nowadays. Based on \cite{4,6,10–12,19,22,25}, we offer the following typology:

- methods based on convolution criteria into a single one;
- methods based on imposing constraints on the criteria;
- target programming;
- methods based on finding a compromise solution;
- methods based on man-machine procedures decision making (interactive programming).

There are other ways of classifications of the methods, such as based on information available about the importance of criteria \cite{12}, etc.

When solving discrete multiobjective problems, a Pareto set formation requires a brute-force search. Thus, the way is intractable for large dimension problems. Taking into account the specifics of a problem is a very useful approach that enables to reduce the search significantly in many cases \cite{5}.

Majority of methods for constructing a set of effective solutions use certain optimality conditions. Most often, the necessary conditions are applied such as if a point is effective (in one or another sense, for example, according to one of the evaluation criteria \cite{6–9}), then it is a solution to the problem (possibly with some additional constraints) of optimizing function of a special form with properly assigned parameters involving in this function and (or) constraints. Such a replacement of MP by a parametric family of standard optimization problems is called a scalarization of the original problem \cite{22,25}. If the optimality conditions are sufficient, then a set of solutions of the parametric problem is its effective solutions’ set. On contrary, a feasible set, constructed by means of scalarization, may contain extra points that should be identified and eliminated since the scalarization problem is, in most cases, is a relaxation to the original one. Thus, most common MO methods are the method of reducing MP to single-objective by convolution of a vector criterion into a super-criterion (further referred to as CM), the method of priorities, and their generalization - the method of successive concessions (further referred to as SCM) \cite{10,22,25}. First one reduces the MP to a single-objective one, other two - to a sequence of single-objective problems.

In CM, a weighted sum of $F(X)$-components is considered as in a super criterion:

$$\Phi(x) = \sum_{l=1}^{L} \alpha_l f_l (x), \alpha_l \leq 0, \ l \in J_L, \ \alpha_l \geq 0, \ l \in J_L \setminus J_L', \ \sum_{l=1}^{L} |\alpha_l| = 1, \quad (11)$$

and then the following problem is solved:

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

When performing the convolution \cite{11}, the main issue and is a right choice of the coefficients $\alpha_l, \ l \in J_L$, the relative importance of the criteria implying that a solution $x^*$ of \cite{3} coincide with a solution $x'$ of \cite{12,10,23}.

By SCM \cite{22}, the individual criteria are ordered with respect to their relative importance - $f_{i_1} \geq f_{i_2} \geq \ldots \geq f_{i_L}$. Then the first, most important criterion, is maximized, and a constraint
A particular case of MO on $\mathcal{C}$-set $B_n$ of $n$-dimensional Boolean vectors or the general $\mathcal{C}_b$-set of permutations $E_{nk}(G)$ induced by $n$-element numerical multiset $G = \{g_1, \ldots, g_n\}$, $g_1 \leq \ldots \leq g_n$, containing $k$ different elements. A particular case of MO on $E_{nk}(G)$ considered by now corresponds to $k = n$, where $G$ is a set. It is $E_n(G)$ called $\mathcal{C}_b$-set of permutations without repetitions or simply $\mathcal{C}_b$-set of permutations. An interesting feature of such ECOPs is that $E'$, and accordingly $X$, coincides with the set of vertices of its convex hull:

$$E' = \text{vert } P', \quad P' = \text{conv } E'.$$

Thus, $E'$ is vertex-located. Also, both of these sets are inscribed into hyperspheres - $\exists x^0 \in \mathbb{R}^n, \exists r \in \mathbb{R}^+ : (x - x^0)_E^2 = r^2$, $E \in \{E_{nk}(G), B_n\}$. (15)

This means, they both are spherically-located.

From this, it follows that MECOP (3), (15) allows reformulating in the form of a continuous MP as follows (15, 29):

$$F(x) \rightarrow \max, \quad x \in P' \cap S_r(x^0), \quad h_i(x) \leq 0, \quad i \in J_m,$$

where $P'$, $S_r(x^0)$ are represented in an analytic form, $h_i(x) \leq 0, \quad i \in J_m$, - are constraints that single out $X$ from $E'$.

Respectively, the MCP is reformulated as a multiobjective global optimization problem. It can be solved by means of nonlinear optimization. Moreover, (16), (17) allows assuming that...
$F(x), h_i(x), i \in J_m$, are convex \[18,29,31\], thus, convex optimization techniques are applicable to such MECOP as well.

Another equivalent formulation of MECOP is related to the possibility of representing it as a problem on vertices of a geometric graph $\mathcal{G} = (E', E)$, $E = \text{edges } \mathbf{P}'$, which is a skeleton graph of $\mathbf{P}'$ and has the form (16),

$$x \in E' \subset \mathbb{R}^n, h_i(x) \leq 0, i \in J_m.$$ 

To this formulation, different graph techniques become applicable \[1-3\], which can be combined with properties of the Euclidean space \[26\]. Finally, the graph $\mathcal{G}$ can be complemented by additional edges decreasing diameter of the $\mathcal{G}^* = (E', E^*)$ (further referred to as an auxiliary graph), $E^* \supset E$. In turn, forming a directed graph from $\mathcal{G}^*$ based on a specifics of ECOP allows applying techniques from \[2,3\].

Further, we focus on the case $E' = E_{nk}(G)$, thus, the problem (16),

$$x \in E_{nk}(G), h_i(x) \leq 0, i \in J_m$$

(further referred to as MCPP) will be considered, and the corresponding ECOP (12), (18) (further referred to as CPP). Consequently, the constraint problem on the generalized $C_\ell$-set of permutations is considered.

CPP is a discrete optimization problem solvable by relevant methods such as branch-and-bound, cutting, branch and cuttings, etc. \[9,21\]. Here should also be noted the method of directed structuring CLM described in \[1-3\] are highly promising for solving such a class of problems.

We study the vector optimization problems on a combinatorial set of permutations and offer a method for solving such problems using Graph Theory, which take into account structural and geometric properties the set of permutations.

**The Modified Coordinate Localization Method (MCLM)**

CLM was original developed to solve a feasibility problem - to find $x \in E_n(G)$ satisfying equality $f(x) = a^T x - b = 0$ \[1,3\]. It uses a reformulation of the problem as CPP (further referred as CPP1):

$$f(x) \rightarrow \max; x \in E_{nk}(G); f(x) \leq 0.$$ 

For CPP1, CLM uses its representation in the form of a directed graph, where the direction of arcs corresponds to the increasing values of the objective function. An important issue is that the usage of above reformulation allows performing a search within a feasible domain of CPP1 - $D = \{x \in E_{nk}(G); f(x) \leq 0\}$ and move towards its solution from a minimizer of $f$ to a minimizer.

When implementing the method, branching and estimate evaluating, based on the examining a feasible domain $X$ as a subset $E_{nk}(G)$, are performed allowing to prune a part of the branches.

To a particular CPP, a structural graph $\mathcal{G}'$ (further referred to as SG) is associated, which is constructed from the auxiliary graph $\mathcal{G}^*$. In turn, $\mathcal{G}^*$ is formed on the basis of a skeleton graph $\mathcal{G}$ of the generalized permutohedron $\mathbf{P}' = P_{nk}(G)$. Its important feature of SG is that, when solving an optimization problem, the only a part of vertices associated with elements of $X$ is analyzed, thus, avoiding a brute-force search.

A skeleton graph $\mathcal{G}$ of $\mathbf{P}'$ has edges $(u, v)$ whenever $u, v \in E' = E_{nk}(G)$ are differ by an adjacent transposition. It is a subgraph of $\mathcal{G}^*$, for which $u, v \in E'$ are adjacent whenever $u, v \in E'$ are differ...
by a transposition. A necessity to consider the auxiliary graph instead of \( G \) is caused by our goal to solve MPP examining a part of the graph vertex set and moving along its edges. To understand the difference between \( G, G^* \), let us compare their degrees of vertices. \( d_G = \sum_{i=1}^{k-1} n_i n_{i+1} \), where \( n_i \) is a multiplicity of \( e_i, i \in J_{k-1} \), \( S(G) = \{e_1, ..., e_k\} \) is a ground set of \( G \), \( e_1 < ... < e_k \). At the same time, \( d_{G^*} = \sum_{1 \leq i < j \leq k} n_i n_{i+1} \).

The auxiliary graph \( G^* \) can be constructed by applying a series of the following method of generating \( E' \) and forming \( E \) from the obtained Hamiltonian paths.

**A scheme of a recursive construction method of \( G^* \):**

1. Set \( E = \emptyset, \kappa = k \);
2. Start from the point \( x = g = (g_{i_1}, ..., g_{i_n}, e_\kappa) \in E' \). Assign \( i = n \).
3. Fix an item with an index \( i \) in the sequence of coordinates of the current point \( x \). The rest \( i - 1 \) elements are generated as follows:
   
   (a) moving from left to right, find the lowest position where \( g_i > g_{i+1} \);
   
   (b) look for the smallest element \( g_j \) located on the left and smaller than it;
   
   (c) make a transposition of elements \( g_j \) and \( g_i \) getting \( y \in E' \);
   
   (d) \( E = E \cup \{x, y\} \);
4. Assign \( x = y, i = i - 1 \). If \( i \geq 1 \), go to Step 3.
5. Set \( \kappa = \kappa - 1 \). If \( \kappa \geq 1 \), go to Step 2.

**Remark 1.** If \( G = J_n \), \( E' \) turns into a standard set of permutations \( E_n = E_n(G) \). Further, we mostly consider this case.

![Figure 1. Graph \( G \) \((n = 3)\)](image)

![Figure 2. Graph \( A \) \((n = 4)\)](image)

![Figure 3. Graph \( B \) \((n = 4)\)](image)

Now, consider in details a structure of the auxiliary graphs \( G^*_n = G^* \) for \( n = 3, 4 \), \( E' = E_n \). Fig. [1] depicts the graph \( G^*_3 \). For \( n = 4 \), four graphs \( A, B, C, D \) isomorphic to \( G^*_3 \) is associated. Therefore, they can be formed from \( G^*_3 \). For instance, the subgraph \( A \) with \( x_4 = 4 \) is presented in Fig. [2] and formed from \( G^*_3 \) by adding 4 to the labels on the right. The subgraph \( B \) corresponds to \( x_4 = 3 \) and is shown in Fig. [3]. It can be obtained from Fig. [2] by substitutions 3 \( \rightarrow \) 4, 4 \( \rightarrow \) 3 in the labels. Similarly, the remaining graphs \( C, D \) are constructed fixing 2 and 1 on the last position, respectively.

68
Note that in the subgraphs of $G^*_4$, the place of the last element is uniquely determined. In general, for $G^*_n$, all the subgraphs $G_i$, $i \in J_n$, are copies (projections) of subgraph $G^*_n$, and they can be ordered by the value of the last coordinate of its vertices. In turn, $G_i$, $i \in J_n$, are connected by edges of other partitions of $E'$ by $n$ subgraphs corresponding to fixing coordinates other than $x_n$.

In the general case of $E' = E_{nk}(G)$, the partition will be made by $k$ subgraphs corresponding to a fixed coordinate $i \in J_n$. In general, these subgraphs are not isomorphic, and their structure is completely determined by multiplicities of inducing multisets \[26\].

This partition of $G^*$ provides the possibility of constructing special graph methods of combinatorial optimization on the set $E_n(G)$, which was used in studies \[23\].

Let CPP2 be the following single-objective constraint program

$$f(x) \to \text{max}; \; x \in E' = E_{nk}(G); \; h_i(x) \leq 0, \; i \in J_m,$$

and $\langle x^*, f^* \rangle$ - be its solution.

**Definition 1.** A directed graph $G' = (V', E')$ is called a structural graph (SG) of CPP1, if it satisfies the following conditions: a) $X = V'$; b) $E' \subseteq E^*$; $\forall x, y \in X$, if $\{x, y\} \in E^*$, then: a) if $f(x) \leq f(y)$, then an arc $(x, y) \in E'$; b) if $f(x) \geq f(y)$, then $(y, x) \in E'$.

Thus, SG $G'$ is a graph of the feasible domain of CPP2, where directions of arcs correspond to increasing of $f(x)$.

Let the following linear CPP2 (further referred to as LCPP) is considered:

$$f(x) = a^T x \to \text{max}; \quad (19)$$

$$h_i(x) = a_i^T x - b_i \leq 0, \; i \in J_m; \quad (20)$$

where $a_1 \geq a_2 \geq ... \geq a_n$, \(21\)

$$x^{min} = \arg\min_{x \in E'} f(x) \in X. \quad (22)$$

Now we present a generalization of CLM for solving CPP2. CLM was developed for solving a specific problem of this class with an equality-constraint \[13\]. For the generalization of CPP1, we add the condition \[22\], and it is the only restriction in comparison with a general LCP on sets of this class. It is added in order to use the same idea as CLM, namely, performing a search within a feasible domain. According to \[21\] and well-known properties of linear function on $E_{nk}(G)$, \[22\] can be rewritten in the form

$$x^{min} = g = (g_1, ..., g_n) \in X. \quad (23)$$

LCPP is solved to optimality, if

$$x^{max} = \arg\max_{x \in E'} f(x) = (g_n, ..., g_1), \quad (24)$$

wherefrom $x^* = x^{max}$. In case if \[24\] does not hold, we organize a directed search of $x^*$ on $G'$ based on the decomposition of $G'$ into directed subgraphs, similar to those presented earlier for $G^*$.

Let us fix $Q \leq n - 3$ and choose $\Lambda = (\lambda_q)_{q \in J_Q} \subset J_n$. To $\Lambda$, a graph $G'(\Lambda)$, which is an induced subgraph of $G'$, is associated, such as

$$G'(\Lambda) = (X(\Lambda), E(\Lambda)) \subseteq G': X(\Lambda) = \{x \in X : x_{n-q+1} = g_{\lambda_q}, \; q \in J_Q\}. \quad (25)$$
Similarly, induced subgraphs \( G(\Lambda) \subseteq G \), \( G^*(\Lambda) = (V(\Lambda), E^*(\Lambda)) \subseteq G^* \), where \( V(\Lambda) = \{x \in E' : x_{n-q+1} = g_\lambda, q \in J_Q\} \) can be defined by eliminating from their vertex sets those vertices, who do not satisfy the constraints:

\[
x_{n-q+1} = g_\lambda, q \in J_Q. \tag{26}
\]

Consider a set \( \Psi = \{\Lambda \subset J_n : |\Lambda| = Q\} \). It is a cover of \( J_n \) by \( A^Q_n \) subsets, where \( A^Q_n \) is the number of \( Q \)-permutations out of \( n \). The cover \( \Psi \) induces a partition of \( G \) into \( A^Q_n \) subgraphs of type \( \ref{25} \). A structure of a permutation graph \( G \) is such that, for any \( \Lambda \in \Psi \), a graph \( G(\Lambda) \) is a permutation graph of \( E'(\Lambda) \) as well, and its dimension is \( n - Q \). Here,

\[
E'(\Lambda) = E_{n-Q,k(\Lambda)}(G(\Lambda)), \quad G(\Lambda) = G\{g_\lambda\}_{q \in J_Q},
\]

\( k(\Lambda) \) is a cardinality of a ground set of \( G(\Lambda) \). In particular, if \( E' = E_n(G) \), then \( E(\Lambda) = E_{n-Q,k-1}(G(\Lambda)) \), thus, all the subgraphs are isomorphic to \( G_{n-Q} \). The same holds for \( G^* \). Respectively, \( G'(\Lambda) \) are structural graph of the corresponding LCPP (further referred as LCPP(\( \Lambda \)) - \( \ref{19}, \ref{21}, \ref{26} \). LCPP is a particular case of LCPP(\( \Lambda \)), namely, LCPP(\( \emptyset \))=LCPP(\( \Lambda \)). So, for arbitrary LCPP(\( \Lambda \)), \( \ref{23} \) and \( \ref{24} \) are generalized as follows:

\[
x^{\text{min}}(\Lambda) = \arg \min_{x \in E'(\Lambda)} f(x) = (g_{\mu_1}, \ldots, g_{\mu_{n-Q}}, g_{\lambda_Q}, \ldots, g_{\lambda_2}, g_{\lambda_1}); \tag{27}
\]

\[
x^{\text{max}}(\Lambda) = \arg \min_{x \in E'(\Lambda)} f(x) = (g_{\mu_{n-Q}}, \ldots, g_1, g_{\lambda_Q}, \ldots, g_{\lambda_2}, g_{\lambda_1}), \tag{28}
\]

where \( \{\mu_i\}_{i \in J_{n-Q}} = J_n \setminus \{g_\mu \leq \ldots \leq g_{\mu_{n-Q}}\} \).

A fixation of \( Q \), which does not depend on \( n \), induces a partition of \( G' \) into a polynomial number of subgraphs \( \Omega = \{G'(\Lambda)\}_{\Lambda \in \Psi} \), similar to the original one. They induce branches of a search tree of LCPP. Now, if a search of \( x^* \) is organized in such a way that for examining \( \Omega \)-components, it is sufficient to consider a polynomial number of points, the problem will be polynomially solvable.

The search will be organized as follows.

**MCLP for LCPP2**

**Input:** LCPP2.

**Output:** \( (x^*, z^*) \).

**An algorithm:**

- **Step 1.** Choose \( Q \), form the family \( \Psi \). Set \( x^* = g, z^* = c^T g; \)
- **Step 2.** For each \( \Lambda \in \Psi \), verify:
  - if \( x^{\text{min}}(\Lambda) \) does not satisfy \( \ref{20} \), then LCPP(\( \Lambda \)) is infeasible;
  - if \( x^{\text{max}}(\Lambda) \) satisfy \( \ref{20} \), it is a solution of LCPP(\( \Lambda \)). If \( z^* < z^{\text{max}}(\Lambda) = c^Tx^{\text{max}}(\Lambda) \), then set \( x^* = x^{\text{max}}(\Lambda), z^* = z^{\text{max}}(\Lambda) \);
  - otherwise, apply a search within a family of subgraphs-grids (see Algorithm 1), choosing as a starting point \( x^{\text{min}}(\Lambda) \in X \) according to \( \ref{27} \).

**Definition 2.** For \( E = E_n(G) \), a grid \( Gr(\Lambda) \) of a subgraph of \( G'(\Lambda) \) is a grid of the dimension \( (n - Q - 1) \times (n - Q) \) with the following properties:

- **its node set** \( \text{Nodes}(\Lambda) = \{p_{ij}\}_{i \in J_{n-Q-1}, j \in J_{n-Q}} \subseteq E' \); 
- **its top-left node** (a source) is \( \ref{27} \) - \( p_{1,1} = x^{\text{min}}(\Lambda) \);
• last $Q$ coordinates are fixed;
• from left to right, $n - Q$-th coordinate of nodes decreases gradually from $g_{\mu_{n-Q}}$ to $g_{\mu_1}$, and first $n - Q - 1$ coordinates are ordered decreasingly;
• from top to bottom, $n - Q$-th coordinate of nodes is fixed, $n - Q - 1$-th coordinate decreases gradually from its original value to $g_{\mu_1}$, and the first $n - Q - 2$ coordinates are ordered decreasingly.

As a result, a bottom-right node (a sink) $p_{n-Q-1,n-Q}$ will have $g_{\mu_2}, g_{\mu_1}$ on last two positions, respectively.

Let us introduce a lexicographic order on the grid’s nodes in the following way:

$$\forall p_{ij}, p_{i'j'} \in \text{Nodes}(\Lambda) \quad p_{ij} \preceq p_{i'j'} \iff f(p_{ij}) \leq f(p_{i'j'})$$

and formulate an important statement in terms of the lexicographic order.

**Theorem 1.** If $\Lambda \in \Psi$ then for $Gr(\Lambda)$ it is true:

$$\forall i \in J_{n-Q-2}, j \in J_{n-Q} \quad p_{ij} \preceq p_{i+1,j}, p_{ij}, p_{i+1,j} \text{ are differ by a transposition};$$

$$\forall i, j \in J_{n-Q-1} \quad p_{ij} \preceq p_{i,j+1}; p_{ij}, p_{i,j+1} \text{ are differ by a transposition}.$$ 

**Corollary 1.**

$$\forall i \in J_{n-Q-2}, j \in J_{n-Q} : (p_{ij}, p_{i+1,j}) \in E^*(\Lambda); \text{ if } p_{ij}, p_{i+1,j} \in X, \text{ then } (p_{ij}, p_{i+1,j}) \in E(\Lambda);$$

$$\forall i, j \in J_{n-Q-1}, (p_{ij}, p_{i,j+1}) \in E^*(\Lambda); \text{ if } p_{ij}, p_{i,j+1} \in X, \text{ then } (p_{ij}, p_{i,j+1}) \in E(\Lambda);$$

**Corollary 2.** In a sink of $Gr(\Lambda)$, it is attained a maximum of $f$ on the grid:

$$f(p_{n-Q-1,n-Q}) = \max_{p_{ij} \in \text{Nodes}(\Lambda)} f(p_{ij}).$$

Theorem 1 means that we perform a directed search of $x^*$ examining $\text{Nodes}(\Lambda)$ from left to right and from top to bottom. Corollary 1 implies that neighboring nodes in $Gr(\Lambda)$ are adjacent in $G^*$, and the direction of the arcs is uniquely determined by indexes of the nodes. Finally, Corollary 2 says that for examining $Gr(\Lambda)$ it is sufficient to check the only $p_{n-Q-1,n-Q}$, if this node is in $X$.

To decrease the number of nodes examined, the following statement can be applied.

**Theorem 2.** If $\Lambda \in \Psi$ then for $Gr(\Lambda)$-nodes it is true:

$$\forall i \in J_{n-Q-1}, \text{ if } \exists j' \in J_{n-Q-1} \quad p_{ij'} \in X, p_{i,j'+1} \notin X, \text{ then } p_{ij,j'+1} \notin X, j > j' + 1;$$

$$\forall i, j \in J_{n-Q-1}, \text{ if } \exists i' \in J_{n-Q-2} \quad p_{i'j} \in X, p_{i'+2,j} \notin X, \text{ then } p_{i,j} \notin X, i > i' + 1.$$ 

Let

$$x^\min(Gr(\Lambda)) = \arg \min_{x \in E'(\Lambda) \cap Gr(\Lambda)} f(x),$$

$$x^\max(Gr(\Lambda)) = \arg \max_{x \in E'(\Lambda) \cap Gr(\Lambda)} f(x).$$

By construction, $x^\min(Gr(\Lambda)) \in X$, hence, by (27),

$$x^\min(Gr(\Lambda)) = x^\min(\Lambda) = (g_{\mu_1}, \ldots, g_{\mu_{n-Q}}, g_{\lambda_Q}, \ldots, g_{\lambda_2}, g_{\lambda_1}).$$

At the same time,

$$z^\max(Gr(\Lambda)) = f(x^\max(Gr(\Lambda)) \geq f(x^\min(\Lambda)), \quad (29)$$

where $x^\min(\Lambda)$ is given by (28).
Algorithm 1 of examining $Gr(\Lambda)$ and a search of $x^{*'} = x^{\max}(Gr(\Lambda))$, $z^{*'} = f(x^{*'})$.

- Step 1. Set $x^{*'} = x^{\min}(\Lambda)$, $z^{*'} = f(x^{*'})$;
- Step 2. Check $x^{\max}(Gr(\Lambda)) \in X$, then $x^{*'} = x^{\max}(Gr(\Lambda))$, $z^{*'} = z^{\max}(Gr(\Lambda))$, finish;
- Step 3. Moving along rows from left to right or from top to bottom, examine a part of Nodes(\Lambda) according to Theorem 2, and improving $z^{*'}$ and updating $x^{*'}$ as far as new nodes in $X$ are found.

In Fig. 4, it is shown a grid $Gr_1 = Gr(\Lambda)$ for $G = J_{n}, n = 6, Q = 1, \Lambda = \{6\}$. Here, a source is $g = (1, 2, 3, 4, 5, 6, 7)$, and a sink is $x^{\max}(Gr(\Lambda)) = (3, 4, 5, 2, 1, 6)$, unfeasible nodes of the grid is shadowed. It is clear that, here, (29) is fulfilled as a strict inequality, and $x^{\max}(Gr(\Lambda)) \neq x^{\max}(\Lambda) = (5, 4, 3, 2, 1)$. Thus, the only partial examination of $G'(\Lambda)$ is performed by $Gr(\Lambda)$. On the other hand, Figure 5 depicts a grid $Gr_2 = Gr(\Lambda)$ for $G = J_{n}, n = 3, Q = 0, \Lambda = \emptyset$, where $x^{\max}(Gr(\Lambda)) = x^{\max}(\Lambda) = (3, 2, 1)$. Now, examining $Gr(\Lambda)$ is equivalent to the one of $G'(\Lambda)$.

In Fig. 6 it is shown a directed graph $G^{\ast}_{3}^{\ast}$, which is isomorphic to $G^{\ast}_{2}^{\ast}$ and is associated to $Gr_2$, wherefrom it is seen, why a $G$-consideration is nor sufficient for the subgraph-grid approach to LCPP2.

Thus, only provided the choice $Q = n - 3$ (cases $n = 1, 2$ are evident), MCLM allows obtaining $x^{*}$, while requires examining a non-polynomial number of subgraphs and grids, namely, $A_{n}^{n-3} = n!/3$. At the same time, choosing a fixed $Q$, the scheme allows obtaining an approximate solution of CPP2 in polynomial time on $n$. Depending on accuracy, we wish to achieve and time available, $Q$ can be chosen closer to $n - 3$ or to 0.

Thus, presenting MLCM, we offer a flexible approach to solve LCPP2 including as greedy algorithms for $Q < n - 3$, as well as a brute force search type for $Q = n - 3$.

Remark 2. If MLCM is applied to LCPP on $E_{nk}'(G)$, where $k < n$, the subgraph-grids are rectangular as well. However, some adjacent nodes of the grids may correspond to the same point of $E'$. Also, the grid dimensions depend on $Q, n, K, \Lambda$ and do not exceed $(n - Q - 1) \times (n - Q)$.
MLCP in Multiobjective Optimization on $E_{nk}(G)$

Let MP \[2\] be linear given in the form

$$ f_l(x) = a_l^T x \rightarrow \max, l \in J_L; $$

s.t. \[20\], $x \in E' = E_{nk}(G)$

(further referred to as a multiobjective LCPP, MLCPP). If $m = 0$, hence \[20\] is absent, it is an unconstrained MLCPP, otherwise, - a constrained MLCPP.

Choosing coefficients of the relative importance of criteria \[30\] and applying a convolution to them in accordance to \[11\], \[12\], we get a linear CPP with $f(x) = \Phi(x)$. If, in addition, \[22\] holds, it is LCPP, to which MLCP is applicable directly. In particular, unconstrained MLCPPs will always be solved on the second stage of MLCP.

Step 2.

To unconstrained MLCPP, MLCP can be used in SCM as well. For that, in \[14\], values of concession $\Delta_l \geq 0, l \in J_{L-1}$, need to be chosen such that to presume fulfillment of conditions, similar to \[22\] on each iteration. Namely,

$$ x^{l,min} = \arg\min_{x \in E'} f_l(x) \in X^{l-1}, l \in J_{L-1}. $$

Conclusions and Future Research

In the paper, the modified coordinate localization method (MLCM) is presented. It generalizes the coordinate localization method (CLM) to the whole class of constrained linear programs on sets of Euclidean configurations of permutations, to an arbitrary finite number of inequality constraints, and with a minor restriction. Two approaches to multiobjective linear optimization on these combinatorial sets are described, implementing MLCM, the convolution and successive concessions methods to multicriteria optimization on permutations, Boolean permutations, and other classes of multi-permutations.

MLCM and the multiobjective approaches may be extended and generalized to solve multiobjective problems on other classes vertex-located $C$-sets [13,15,16,26,28].
References