Assessment of aircraft prospects using a combined method of identifying preferences

V Sudakov1,* and T Sivakova2

1Department of Mathematical Cybernetics, Moscow Aviation Institute (National Research University), 4 Volokolamskoe Highway, 125993, Moscow, Russian Federation
2Department of Problems of Mathematical Modeling and High Performance Computing, Keldysh Institute of Applied Mathematics, 4 Miusskaya Square, 125047, Moscow, Russian Federation

*E-mail: SudakovVA@mai.ru

Abstract. Recently, the task of replacing the aircraft fleet has been in demand. The aim of the work is a multi-criteria assessment of aircraft and further decisions on their use. To solve this problem, it is proposed to use a combined method for identifying preferences, which is well algorithmic and allows you to take into account dependences on preferences in the multidimensional system of values of the decision maker. An algorithm based on the method of enumeration on a vector lattice has been developed. This algorithm helps the decision maker to take into account a large number of criteria and decide on the need to replace the aircraft. The advantages include a reduction in the calculation time. An interface for entering editing and preferences is proposed, which makes it possible to assess the feasibility of introducing new and withdrawing old aircraft.

1. Introduction
Over the past years, there has been a significant increase in air traffic flows for both passenger and freight transport [1,2]. In this regard, one of the urgent tasks is the modernization of the aircraft fleet. The main factors determining the demand for new aircraft are the development of air transportation and the disposal or replacement of the existing fleet.

The demand for passenger air transportation, in turn, depends on many factors, but, first of all, on the size and rate of development of the gross domestic product (GDP), population and the level of development of the air transportation industry and alternative modes of transport. Understanding market needs, in particular the key reasons for renewing the commercial aircraft fleet in operation, is a critical factor in developing a new product strategy.

Airlines often have to choose whether to use an old aircraft before the resource is exhausted or buy a new one, while it is necessary to take into account such criteria as flight safety, compliance with environmental standards, fuel efficiency, etc. But also the presence of dependencies between the criteria for the preferences of the decision maker (DM). In addition, one has to take into account the fact that often the DM does not have experience in carrying out specific work.

Currently, work in the field of a decision support system (DSS) for finding the best is possible only for well-studied, fairly simple tasks, but in practice, semi-structured problems are more common for which no fully formalized decision-making algorithms have been developed. Therefore, it is in demand in the DSS for practical human tasks to solve non-formalized tasks using formal methods and computer modeling, such as decision modeling methods, decision support systems, expert systems, etc. When
solving such problems, optimization methods have proven themselves well [3,4], which are in great demand in the field of modern computational and applied mathematics, but require rather large expenditures of computational resources.

Optimization methods have been developed since the 19th century; they were the first thoroughly studied problems of finding the extremum of functions in the presence of inequality-type constraints [5]. The main method of directed enumeration of adjacent vertices in the direction of increasing the objective function is the simplex method, which has become the main one in solving linear programming problems [6].

Simultaneously with the development of linear programming, much attention was paid to nonlinear programming problems in which either the objective function or constraints, or both are nonlinear. In 1951, Kuhn and Tucker published a paper, which provides necessary and sufficient optimality conditions for solving nonlinear programming problems. This work served as the basis for further research in this area [7].

In the problems that one has to face in practice, the objective function depends on many parameters and has a high dimension, which complicates the use of the methods described above. To eliminate this, it is proposed to divide it into multistage processes with low dimensionality for each one [8].

In recent years, the following algorithms have been in demand: tabu search, simulated annealing, greedy, genetic and swarm intelligence algorithm [9-11].

The difficulty of solving the optimization problem is determined by the number of its variables, the type of the objective function, as well as the type and number of constraints. Moreover, often the methods developed for solving one type of problem or another turn out to be useful for solving more complex problems. So, for example, one-dimensional optimization algorithms are widely used in solving multidimensional problems; many conditional optimization algorithms use unconditional optimization methods or are their modification; methods for solving linear programming problems are used to solve nonlinear programming problems, etc.

Among the minimization methods, one can conditionally distinguish finite-step and infinite-step methods [12-14]. Finite-step, or finite, are methods that guarantee finding a solution to a problem in a finite number of steps. Finite-step methods can be constructed only for some special types of optimization problems, for example, linear and quadratic programming problems. For infinitely step methods, the achievement of a solution is guaranteed only to the limit.

Thus, for the methods that allow solving the scheduling tasks most efficiently, it is necessary to set up that takes into account the peculiarities of the tasks.

In relation to the problem of planning aircraft replacement, it is proposed to use the Lemke-Shpilberg algorithm [15,16], which allows to reduce the number of solutions when using the enumeration method. The process of finding a solution, according to this method, is to carry out “steps” forward or backward through the possible options for solving the problem. If the dropout criterion shows that moving forward from the current point does not make sense, then the entire branch of decisions is excluded from the calculation, along which it would be possible to move forward without taking into account the criterion.

It is proposed to use a combined method of identifying preferences, which allows evaluating the effectiveness of aircraft in terms of their entry into the modern air transportation market, taking into account the preferences of the DM.

2. Materials and methods

Using the example of a combined method for identifying preferences for decision support, we will consider how to choose an aircraft. Let each aircraft have a set of top-level criteria:

- specifications;
- efficiency;
- import dependence.

For each criterion of the top level, let us set a scale with gradations:

- technical characteristics: t11 - good (1), t12 - satisfactory (0.5), t13 - unsatisfactory (0);
- efficiency: t21 - good (1), t22 - satisfactory (0.5), t23 - unsatisfactory (0);
- import dependence: t31 - low (0), t32 - high (1).
The way each aircraft evaluates the criteria took the following values:
- No.1: technical characteristics - 0.25, efficiency - 0.8, import dependence - 1;
- No.2: technical characteristics - 0.5, efficiency - 0.35, import dependence - 0.

It is necessary to identify preferences on the sets of gradations of criteria, namely, to determine the preference areas $M_k$ and their preference levels $P(M_k)$. Let's introduce three areas of preference:
- $M_1 = \{t_{12}, t_{13}, t_{22}, t_{23}, t_{32}\}$ - an area with low preference with a preference level $P(M_1) = 2$;
- $M_2 = \{t_{12}, t_{21}, t_{22}, t_{32}\}$ - an area with an average preference with a preference level $P(M_2) = 3$;
- $M_3 = \{t_{12}, t_{13}, t_{22}, t_{23}, t_{31}\}$ - an area with good preference with a preference level $P(M_3) = 4$.

After the stages of determining the cells $T_l$, forming the preference function $Y(X)$ and substituting the values of the criteria of alternatives No.1 and No.2 into the preference function $Y(X)$, we will receive the following result - No.2 is preferable to No.1.

At the moment, there are many decision support systems. However, only a small part of them provide the ability to solve decision-making problems by methods that involve the use of preference functions to obtain results. As a result, the world has very limited experience in designing interfaces for describing preference functions in decision support systems. Such multi-criteria decision support methods can be very useful in completely different areas of research, solving scientific and technical problems, business planning, production, financing an investment project, management accounting, etc. For the development of this direction, an interface for users and operators of multi-criteria decision support systems was designed that meets modern requirements and trends in software design (figure 1).

Figure 1. Preference areas interface screen.

Figure 1 shows the web interface for entering and editing preferences. This interface will fit into the width of the user's web browser. Small margins on the left and right edges will be provided for users with widescreen monitors (for example, with a 16:9 ratio). Each preference area will be half the given width of the entire interface. This number of preference areas in one line is optimal for compactness and normal display of long text fields of the interface.

For the interface for directly describing the preference areas, all the main front-end development tools were used:
- HTML. It was used to create an interface structure in the form of a tree of nodes, where each node is an element or a group of elements. Structural entry of one element into another, for example, the
values of criteria in a certain preference area, is necessary for unambiguous work with all interface elements. HTML also provides standard form elements for development - buttons, checkboxes, etc.

- CSS. The visual design, layout, typography, and colorization of interface elements were almost entirely developed in this language.

- JavaScript. This language was used to enable the Add Preference Region button. This action is considered dynamic and requires special handling. In advance in JavaScript code the structure of interface elements for one preference area is set. When you click on the button after all preference areas in the interface, a new, not yet filled preference area with the necessary interface elements is drawn.

The developed interface for direct description of preference areas has the following advantages:

- equally convenient for PC and mobile users when changing preference areas and their preference levels;
- from the point of view of programming - easy and unambiguous in structure and code writing;
- visual enough for DM, helps to quickly navigate to changing indicators.

3. Result and discussion

The solution algorithm is based on the construction and analysis of a vector lattice [17]. Let us consider the application of the method of implicit enumeration over a vector lattice to linear integer problems of the following form:

$$\min \sum_{j=1}^{n} c[j]x[j]$$

$$\sum_{j=1}^{n} a[i,j]x[j] \leq b[i], (i = 1,m)$$

$$x[j] \leq b[j], (j = 1,n)$$

which is reduced to the form where all $c[j] \geq 0$, and the constraints represent inequalities like “≤”. These conditions are not restrictive, since their fulfillment can always be ensured through appropriate transformations.

In cases when, in the optimization problem, the minimum coefficient in the objective function for the $j$-th variable is negative, then it is necessary to introduce a change of variables:

$$x[j] = h[j] - x'[j]$$

(2)

Obviously, to ensure the equivalence of the transformation, it is necessary to make this change not only in the objective function, but also in all constraints. After such a replacement $c'[j] = -c[j] > 0$, and the constant formed in the objective function must be removed (it does not affect the optimization results).

The formal way to transform constraints like “=” to inequalities is to replace each of them with two corresponding inequalities of the opposite sign. A visual graphical representation of the vector lattice for the combinatorial problem ($h[j]=1$ ($j=1,2,3$)) in the case $n = 3$ at is shown in figure 2.
**Enumeration procedure.** The enumeration of points of a vector lattice is carried out along its routes with the help of steps forward and backward.

A route is any path passing through the vertices of a vector lattice connected by arcs. A step forward is a transition to a higher numbered level, and a step back is a reverse movement. Since each step forward is associated with an increase in the corresponding variable by one, we will say “step forward in the j-th variable”. In order to exclude repeated consideration of points when moving forward, a rule is introduced according to which, from any point $x$: steps forward are considered in variables whose numbers satisfy the following conditions:

$$ j \in J_{vs} = \{ j \in \overline{1,n} : j \geq j_s, \text{if } x[j_s] < b[j_s], \text{else } j > j_s \} $$

(3)

$$ j_s = \max j \{ j \in \overline{1,n}: x[j] > 0 \} $$

(4)

where $j_s = \max j \{ j \in \overline{1,n}: x[j] > 0 \}$

(5)

In accordance with this rule, to form search routes along a vector lattice, only its arcs are used. The points for which $J_{vs} = \emptyset$ will be called finite. After moving to such a point using a step forward, a step backward will definitely follow.

The movement along the vector lattice begins and ends at the zero point. The completion of the enumeration at the zero point occurs when all steps forward from it have already been considered.

Now we will consider the rules, the application of which will allow us to reduce the number of solutions (points) enumerated by the algorithm and, therefore, to make the enumeration implicit.

**Allowed point rule.** The method of implicit enumeration on a vector lattice enumerates mostly inadmissible points: as soon as an admissible point is reached while moving forward, this movement stops and a step is taken back.

The rationale for this rule consists in transforming the problem to the form, where all $c[j] \geq 0$, and the direction of optimization is the minimum. At the same time, moving forward cannot lead to a decrease in the objective function, but, as a rule (with a step forward in a variable for which $c[j] \geq 0$), it leads to its increase, that is, to a deterioration in the sense of the direction of optimization to a minimum.

As a special case, the conclusion follows from this rule: if the zero point is admissible, then it is the optimal solution to the problem.

**Inadmissibility criterion (IC).** This criterion is applied to each point of the vector lattice after the initial hit in it when moving with steps forward. It forms sufficient conditions for the impossibility of achieving any admissible decision when moving forward from a given point. If these conditions are met, then they say that the criterion eliminates the given point (the criterion has worked at this point). In this case, it is necessary to stop moving along the vector lattice using steps forward and take a step back.

Let us define additional transformations of the original problem and new concepts and designations that are necessary for a formalized description of the action of the IC.

The original problem should be reduced to a form where all constraints have the form of an inequality like “$\leq$”:

$$ \sum_{j=1}^{n} a[i,j] x[j] \leq b[i] $$

(6)

A balance variable $i$ is introduced into each constraint $y[i] \geq 0 \ (i = \overline{1,m})$. It turns out the following, equivalent to the original system of restrictions:

$$ \sum_{j=1}^{n} a[i,j] x[j] + y[j] = b[i] ; \ y[i] \geq 0 \ , \ (i = \overline{1,m}) $$

(7)

It should be noted that the balance variables $y[i]$, in contrast to linear programming, are not introduced into the optimization variables, but serve only to check the fulfillment of the constraints. When moving along a vector lattice, an iterative recalculation of balance variables can be organized,
which eliminates the need for direct substitution of the vector of optimization variables into constraints to check their implementation.

Let's consider two sets:

- \( I_x \) – the set of constraints not fulfilled at the point;
- \( J^x_i \) – the set of variable numbers, along which it is still \( x \) possible to take a step forward from the point, and for which the coefficients in the \( i \) - constraint are negative.

The formal definition of these sets is as follows:

\[
I_x = \{ i \in \mathbb{I}, m : y_i[i] < 0 \}
\]

\[
J^x_i = \{ j \in \mathbb{I}, n : j \in J_w ; a[i,j] < 0 \}
\]

(8)

(9)

Obviously, taking a step forward on a variable \( x[j] \) increases the balance sheet variable \( y[i] \).

Taking this into account, the formal rule for the action of the IC is determined as follows: if at the point under \( x \) consideration, during sequential verification of the next non-fulfilling constraint (\( i \in I_x \)), it turns out that:

\[
\sum_{j \in J^x_i} a[i,j] (h[j] - x[j]) > y_i[i]
\]

(10)

where \( \sum (...) = 0 \), if \( J^x_i = \emptyset \).

The rationale for this rule is that, even after taking all possible steps forward in all variables \( j \in J^x_i \), it is impossible to ensure such an increase \( y_i[i] \) in which it becomes non-negative, i.e. it is impossible to achieve the fulfillment of the \( i \) restriction while moving forward.

The criterion for the systematic exclusion of alternatives (CSEA). This criterion is applied to individual steps forward from the considered point after at least one unacceptable solution to the original problem has been found. The essence of his work is that he filters out those steps forward that cannot lead to better (smaller) solutions in terms of the objective function. The action of CSEA is based on the use of a filtering constraint:

\[
\sum_{j=1}^{n} c[j] x[j] < z(x)
\]

(11)

Which after the introduction of the balance variable \( y [0] > 0 \) is converted to the form:

\[
\sum_{j=1}^{n} c[j] x[j] + y[0] = z(x)
\]

(12)

Formal rule of CSEA: a step forward in the \( j \) variable from a point \( x \) is not realized if the following inequality is satisfied:

\[
c[j] \geq y_i[0]
\]

(13)

Once again, we emphasize that the CSEA is included in the work as soon as the first admissible solution of the optimization problem is found during the operation of the NPVR algorithm, or from the very beginning of the algorithm, if a feasible solution can be formed on the basis of a priori information. The right side of the filtering constraint is adjusted every time a new feasible solution is found that is better than the previous one. After this adjustment, the dropout rate of steps forward with the help of CSEA increases even more.

Preferred Variable Criterion (PVC). The purpose of this criterion is to order the steps forward from each invalid point that is not the end point and is not screened out IC. Such a problem arises after the first arrival at such a point with the help of a step forward. The purpose of ordering is to look first at those routes that are most likely to lead to valid solutions. The action of the checkpoint to reduce the search, if the ordering goal is achieved, is implemented through joint work with the CSEA: the sooner the first feasible solution is found, the earlier the CSEA will be involved in the work; when finding then
the best admissible solutions, the action of the filtering restriction on screening out steps forward becomes more stringent.

Formally, the PVC rule applied to possible steps forward from the point \((j \in J_v)\), consists in ordering them in ascending order of values:

\[
\sum_{i \in I} a[i,j] [(h[j] - x[j])]
\]  

The idea of such an ordering is, first of all, to consider routes that do not lead to a strong decrease in the balance variables \(y[i]\) in unfulfilled constraints, but rather lead to their increase (if negative \(a[i,j]\)), i.e. to reduce the set \(I_v\). The action of the checkpoint PVC, in contrast to IC and CSEA, is heuristic in nature and cannot always lead to the desired results, since the choice of the route by the smallest value does not guarantee the priority movement towards admissible solutions.

Let's move directly to the algorithm shown in figure 3.

---

**Figure 3.** Vector lattice enumeration algorithm.
Step 1. \( x = 0 \).
Step 2. If \( (x \in D) \), then \( z^0 = 0, \ x^0 = 0 \), go to step 9.
Step 3. If \( (x \in D) \), then
- calculation of the value of the objective function at the next found admissible point;
- if the filtering constraint has not yet been introduced, then enter it with the right-hand side \( z^0 = z(x) \), commit \( x^0 = x \);
- if the filtering constraint has already been formed and the condition \( z(x) < z^0 \) is satisfied, then redefine its right-hand side \( (z^0 = z(x)) \) and \( x^0 = x \);
- go to step 8;
Step 4. If \( (x \notin D) \), then
- checking the point \( x \) for belonging to the set of end points \( (J_{sv} = \emptyset) \). If the point is the end point on the branch of the vector lattice, then go to step 8;
- check point \( x \) according to the IC criterion. If the point does not pass along the IC, then go to step 8.
Step 5. Arrange possible steps forward from point \( x \) according to the checkpoint criterion PVC.
Step 6. If steps forward from point \( x \) have not been considered yet, then the choice of the next step forward (in the order determined by the PVC criterion), otherwise: if \( x \neq 0 \), then - go to step 8;
if \( x = 0 \), then go to step 9.
Step 7. If the CSEA criterion is included in the work (a feasible solution has already been found), then checking the next step forward by the CSEA criterion:
if the chosen step forward according to the CSEA criterion does not pass, then - go to step 6, otherwise - step forward along the vector lattice, go to step 3.
Step 8. Step back along the vector lattice. Go to step 6.
Step 9. If \( (z^0, x^0) \) is defined, then this is the optimal solution, otherwise \( D = \emptyset \).

4. Conclusions
An interface for entering and editing preferences has been developed to evaluate the efficiency of commissioning new types of aircraft and withdrawing old types of aircraft.
The combined method of identifying preferences proposed by the authors makes it possible to assess the effectiveness of aircraft from the point of view of their introduction into the modern air transportation market, taking into account the preferences of the DM.
The enumeration method used in relation to the plane replacement problem can reduce the number of solutions and thus reduce the computation time of the problem.
To optimize the solution of problems, a heuristic search algorithm on a vector lattice is implemented.
The following properties of this algorithm should be noted:
1. The described algorithm enumerates mostly inadmissible points of the domain of possible solutions to the problem. If the starting point of the search can reasonably be brought closer to the boundaries of the active constraints of the original problem by somehow determining more accurately the possible location of the point of the optimal solution, then the amount of enumeration can be reduced even more significantly.
2. This algorithm can be applied to problems with a nonlinear objective function. The condition for the realization of such a possibility is the monotonic non-decreasing character of this function with respect to the optimization variables when the optimization is directed to the minimum.
The methodology applied in the work allows managers to orient themselves in the current circumstances and make further decisions on the need to replace the aircraft, taking into account the interests of the company and the geography of transportation.
This approach can be used not only in the problems of air transport systems, but also in other areas of scientific and applied problems.
References

[1] Kluge U, Paul A, Cook A and Cristóbal S 2017 Factors influencing European passenger demand for air transport. *Proc. 21th Air Trans. Research Soc. World Conf. 2017* (University of Antwerp Stadscampus, Belgium) pp 1-14 DOI: 10.5281/zenodo.3948183

[2] Chen Z. 2017 Impacts of high-speed rail on domestic air transportation in China. *J. Transp. Geogr.* 62 184 DOI: 10.1016/j.jtrangeo.2017.04.002

[3] Rothlauf F 2011 Optimization method. *Design of Modern Heuristics, Natural Computing Series* (Berlin: Springer ) chapter 3 pp 45-102 DOI: 10.1007/978-3-540-72962-4_3

[4] Karpenko A P 2017 *Modern Algorithmsearch Engine Optimization. Algorithmsinspired by Nature* (Moscow: MSTU BAUMAN) p 448

[5] Nogin V D 2016 *Narrowing the Pareto Set. An Axiomatic Approach* (Moscow: Fizmatlit) p 272

[6] Kuhn H W, Tucker A W 1951 Nonlinear programming. *Proc. of the Second Berkeley Symposium on Math. Statistics and Probability. University of California Press* (Berkeley, California) p. 481-492

[7] Taha H A 2007 *Operation Research. An Introduction* (Upper Saddle River, New Jersey: Prentice Hall) p 820

[8] Lessard L, Recht B and Packard A 2016 Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. on Optim.* 26(1) 57 DOI: 10.1137/15M1009597

[9] Deb K 2014 Multi-objective optimization Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques eds. et al. Second Edition pp 403-450 (New York, Springer ) p. 403-405 DOI: 10.1007/978-1-4614-6940-7_15

[10] Gupta K K and Kumar S 2019 Hesitant probabilistic fuzzy set based time series forecasting method. *Granular Comput.* 4(4) 35 https://doi.org/10.1007/978-981-13-0680-8_4

[11] Pedersen M and Chipperfield A 2010 Simplifying particle swarm optimization. *Appl soft comput 10*(2) 618 DOI: 10.1016/j.asoc.2009.08.029

[12] Lemke C, Salkin H and Spielberg K 1971 Set covering by single branch enumeration with linear programming. *Subproblems Oper. Res.* 19 998

[13] Pichler X and Hu X Quantitative Stability Analysis for Distributionally Robust Optimization with Moment Constraints SIAM. *J. Optim.* 26(3) DOI: 10.1137/15M1038529

[14] Zhang J, Xu H and Zhang L 2017 Quantitative stability analysis of stochastic quasi-variational inequality problems and applications. *Mathem. Program.* 165(1) 433 DOI: 10.1007/s10107-017-1116-9

[15] Ali S, Koenig S and Tambe M 2005 Preprocessing techniques for accelerating the DCOP algorithm ADOPT. *Proc. of the Fourth Int. Joint Conf. on Autonomous Agents and Multiagent Systems* (Utrecht, Netherlands) pp 1041-1048

[16] Kofman A and Henri-Laborder A 1977 Methods and models of operations research (Moscow: Mir) p 432

[17] Brower R C, Weinberg E, Clark M A and Strelchenko A 2018 Multigrid algorithm for staggered lattice fermions. *Physical Review D* 97 (11) 114513 DOI: 10.1103/PhysRevD.97.114513