DOI: http://doi.org/10.25728/cs.2023.6.2

FORMING THE GENERATIONS OF NEW TECHNOLOGICAL PRODUCTS AS A SET COVERING PROBLEM

S.A. Barkalov[1], V.N. Burkov[2], P.N. Kurochka[3], and E.A. Serebryakova[4]

[1,3,4]Voronezh State Technical University, Voronezh, Russia
[2]Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

[1]✉ bsa610@yandex.ru, [2]✉ vlab17@bk.ru, [3]✉ kpn55@ramler.ru, [4]✉ sea-parish@mail.ru

Abstract. The development of any enterprise implies improving its control mechanisms for the manager to make decisions based on the achievements of science rather than intuitive ideas of his (or her) personal experience. It is necessary to improve the model-building process in order to eliminate the coinciding peaks of resource consumption when working on multiple projects. For this purpose, the concept of a generation of new technological products can be adopted: a new product is formed from separate prototypes (operating models), which can serve to determine some features of the project under development. Naturally, it is unreasonable to include the entire model range in the generation of new technological products: one should select the minimum number of prototypes required. This problem belongs to the class of set covering problems: complete covering (when the selected prototypes must possess the entire set of properties possessed by the model series under development) or partial covering (when the selected prototypes must possess only some of these properties). Exact algorithms and approximate heuristic algorithms are presented to solve both problems.

Keywords: placement problem, complete set covering problem, partial set covering problem, innovation lifecycle, generation of new technological products, prototype, properties matrix.

INTRODUCTION

The process of creating and improving new technological products (hereinafter called products for brevity) is a priority direction of Russia's development. Hence, the creation of generations of new products predetermines certain progress in the field of design and technological solutions, based on which the former are developed [1, 2]. Applying new solutions in the design and technological sphere implies obtaining new products with new functional properties. Thus, the new functional properties of a developed product are formed using new design and technological solutions [3, 4].

The creation of new products is a very cost-intensive process requiring significant material, human, and financial resources. Any fruitful idea underlying development works will be applied in the new generations of products forming a model range. The concept of a prototype is therefore widely used in engineering. A prototype is an operating model of the processes occurring in a new product being created.

Note that the problem of resource provision arises in the course of any project. Its potential solution is based on that the resources can be distributed over time during the implementation period of the project, i.e., they are not required all at once. As a rule, an enterprise is engaged in several projects simultaneously; hence, it is necessary to organize the project implementation process so that different projects have noncoinciding peaks of resource consumption [5–7]. A generation of new products should be formed by selecting a minimum number of prototypes to reduce the amount of resources required.

In this case, the problem of forming a generation of new products can be solved using optimization methods with some optimality criteria [7, 8]. In other words, the following question arises immediately: what should be optimized?

1. PROBLEM STATEMENT

As a rule, any fruitful idea has a continuation expressed in the creation of a whole series of new products with new functional properties. What can be used as a prototype to develop the next generation of innovation? Naturally, the selection is based on the principle of an identical scope of application: a tractor carries out one set of works whereas a tank another.

An excellent illustrative example is the development of aircraft technology. Consider TU-104, a Soviet jet passenger airplane (the third largest one in the world). In the first two years of its operation (1956–1958), TU-104 was the only jet passenger airplane in the world after the discontinuation of the British De Havilland Comet in the summer of 1956 and until the introduction of the American Boeing 707 into commercial operation in October 1958. Officially, the airplane had the following modifications: TU-104, TU-104A, and TU-104B. It was produced until 1960 and operated until 1981. In addition to TU-104, the TU-124 airplane was developed, including three modifications (TU-124A, TU-124B, and TU-124B). TU-124A, the most successful modification, was soon transformed into a new type of airplanes (TU-134), which also had 19 modifications. TU-134 was manufactured until 1989 and is still in service.

The same picture can be observed for Western products. For example, the British De Havilland Comet—the world's first jet passenger airliner—had Comet 1, Comet 1A, Comet 1XB, Comet 2X, and 12 other modifications. The airplane was manufactured until 1964 and was in service until 1997.

A passenger airliner must combine speed and capacity to transport as many passengers as possible to the desired destination as quickly as possible. This is the key factor when forming the generations of new products of this type.

The simplest solution to form the model range would be the selection of all the created products as prototypes. But this solution is somewhat redundant. As a rule, individual products have sufficiently close properties, and it seems unreasonable to choose all of them as prototypes. A reasonable approach is to take some N of them. Thus, the first optimization criterion is the minimum number of prototypes selected for developing an innovative product.

Assume that there are N products possessing M properties. To describe the particular properties of different products, we introduce the properties matrix A with the following elements: $a_{ij} = 1$ if product i possesses property j and $a_{ij} = 0$ otherwise.

Thus, the property matrix A consists of zeros and ones. Let the properties matrix be formed using a set of N products ordered by the time of creation; since the properties are linked to the product, the products developed in a later period may have new properties but may lose some old ones. As an example, we recall a classical example, vividly manifested during the Chernobyl disaster elimination: the use of semiconductor elements led to the emergence of completely new properties but the anti-radiation stability of the equipment was lost. Therefore, the matrix A will have a ribboned structure: the elements $a_{ij} = 1$ will be grouped mainly near the main diagonal, forming a kind of "ribbon."

Thus, economic considerations [9] bring to the problem of selecting the minimum number of prototypes to develop a product with a given set of properties.

We write this problem as an integer linear programming problem with the objective function

$$\sum_{i=1}^{N} x_i \to \min \tag{1}$$

and the constraints

$$\begin{gathered} \sum_{i=1}^{N} a_{ij} x_i \ge y_j, \quad j=\overline{1, M}, \\ \sum_{j=1}^{M} y_j = m, \\ x_i \in \{0, 1\}, \quad i=\overline{1, N}, \\ y_j \in \{0, 1\}, \quad j=\overline{1, M}, \end{gathered} \tag{2}$$

with the following notations: x_i is a binary variable equal to 1 if product i is selected as a prototype and to 0 otherwise; m specifies the number of properties to be satisfied for all the selected prototypes; y_i is a binary variable equal to 1 if product i must have this property and to 0 otherwise.

Problem (1), (2) is to determine the minimum number of prototypes with given properties, a basis for the further development of this innovation.

By its nature, the problem under consideration resembles the placement problem of infrastructure objects in a given domain: the role of such objects is played by the existing products, and the domains are the properties to be satisfied for the selected prototypes [9].

In this case, two formulations of the problem are possible:

1. The number of properties to be satisfied for the selected set of prototypes is equal to the total number

of properties characteristic of the entire set of products, i.e.,

$$m = M.$$

2. The number of properties to be satisfied for the selected prototypes is equal to or less than their total number, i.e.,

$$m \le M.$$

In graph theory, problems of the first type belong to the class of complete set covering ones; possible algorithms for solving them were described in [10]. Problems of the second type belong to the class of partial set covering ones. Both types of problems are NP-hard.

2. AN ALGORITHM FOR SOLVING THE COMPLETE SET COVERING PROBLEM

This problem is formally described by the objective function (1) and the constraints (2) without the second one. In other words, the objective function (1) remains unchanged, and the system of constraints is written as follows:

$$\sum_{i=1}^{N} a_{ij} x_i \ge 1, \quad j = \overline{1, M}, \qquad x_i \in \{0, 1\}, \quad i = \overline{1, N}. \tag{3}$$

Problem (1), (3) belongs to the class of integer linear programming problems, and the simplex method turns out to be inapplicable here.

The system of inequalities in (3) expresses the requirement that each property is satisfied for at least one product.

If only one product possesses some property, such a product will be called unique.

Proposition 1. *Unique products must be included in the set of selected prototypes.*

P r o o f. Only a unique product has a particular property; no other product possesses that property. Hence, it is necessary to include this product in the set of selected prototypes to cover all properties. ♦

Unique products are easy to determine: it suffices to calculate the sums of all columns of the properties matrix. If the sum of a column equals 1, then the product corresponding to this property is unique and must be included in the solution.

Multiplying all inequalities included in the constraints, i.e.,

$$\prod_{j=1}^{M} \sum_{i=1}^{N} a_{ij} x_i \ge 1, \tag{4}$$

yields a Boolean polynomial of degree M after removing the brackets. In this case, we can formulate the following result.

Proposition 2. *Each term in the Boolean polynomial of degree M of the expanded expression* (4) *represents an admissible solution of the problem satisfying the constraints* (3) *but is generally not the optimal solution.*

P r o o f. The desired solution must satisfy the system of unstrict inequalities (3). (Each property must be satisfied for at least one product.) If a product with this property is single, the corresponding constraint will hold as equality and the product will be unique. Therefore, the entire system of constraints—the unstrict inequalities (3)—can be replaced by a single constraint when multiplying the cofactors for the set of products with the property in question. There may exist several such products. If none of the products possesses a certain property, the corresponding cofactor will contain only zeros and will be equal to 0; therefore, the entire expression will be equal to 0, which violates the constraints. Indeed, the expression (4) is a product with M cofactors (the number of properties possessed by all products). In turn, each cofactor is a kind of list of products that have this property. If we expand the polynomial (4), each term will have degree M and describe an admissible solution. ♦

The trivial solution is to select the entire set of products as prototypes, i.e., all N products. However, the number of candidates for a new generation of innovations can be generally reduced. This is possible if some product partially satisfies the properties that are inherent in some other products. (In this case, they need not to be selected.) This will decrease the number of prototypes selected for a generation of new products. Note that at the stage of data preparation, the problem dimension can be reduced by using the concept of a unique product, which has properties absent in other products. As a rule, such products are those of the latest development. Naturally, such products must be included in the solution.

Thus, the problem is to select the minimum number of prototypes possessing all the properties corresponding to a given model range. There may exist several such sets. All of them are selected and presented to the decision maker, who determines an acceptable solution.

The minimum value of the objective function will be achieved when all constraints in the form of nonstrict inequalities hold as equalities. In other words, each property is implemented for one product only. But this situation is usually not the case in practice: each property is often characteristic of several products. As a result, the constraints (3) will be implemented in the form of strict inequalities [10–12]. Due to the binary character of the variables x_i, we can replace the system of inequalities with a recurrent system of Boolean equations. To be more precise, the following result is valid.

Proposition 3. *Problem* (1), (3) *is equivalent to the following sequence of Boolean equations*:

$$\prod_{j=1}^{M}\sum_{i=1}^{N} a_{ij}x_i = k,\ k = 1, 2, \ldots, l\ , \qquad (5)$$

Here, theoretically, the upper bound for l will be the value $l = N^M$, i.e., the admissible solution in which all N products have all M properties. But this solution is trivial.

P r o o f. The inequality sign in the constraint (3) means that, in principle, several products may possess the same property and the expression (3) will accordingly hold as a strict inequality. Considering the integer nature of the problem, we can replace inequality (4) with the sequence of equalities (5). Expanding the expression (5) yields a Boolean polynomial of degree M. Each such term describes an admissible solution; therefore, to minimize the number of selected products, it is necessary to choose the polynomial term with the smallest number of cofactors but of the maximum degree. ♦

By sequentially solving equation (5) for different k, we arrive at the desired solution in a finite number of iterations.

To solve the problem, it is necessary to write the expression (5) as a polynomial of degree M. A common approach is to choose the most frequently encountered variable in (4): it will have the maximum possible degree after removing the brackets in the Boolean polynomial [10, 13, 14]. In this case, the following result is true.

Proposition 4. *In the Boolean polynomial* (5), *the terms with the minimum number of cofactors correspond to the optimal solution of problem* (1), (3). *In addition, such a term contains the smallest number of the variables x_i, but each of them has the maximum possible degree.*

P r o o f. Each product may have several properties. Hence, by selecting products with the maximum number of properties, one reduces the number of products to be selected. ♦

Based on the properties of Boolean polynomials described in Propositions 1–4, we can develop an exact algorithm for solving the problem. For this purpose, it is necessary to obtain the expanded expression for the polynomial (4). Moreover, the entire set of terms of the expression (4) is not required in explicit form: it suffices to obtain only a few first terms with the minimum number of cofactors of the maximum degree. The degree of each cofactor of the Boolean polynomial (4) must equal the number of properties to be satisfied for the selected prototypes. According to Proposition 4, each such term will be a solution of the problem under consideration.

Note that for a high-dimensional problem, extracting the first terms with the minimum number of cofactors of the maximum degree from the Boolean polynomial is a rather time-consuming and very difficult operation. It has not been computerized so far. Therefore, we propose a heuristic algorithm for solving problem (1), (3), which will be convenient for computer implementation. This algorithm is based on Propositions 1–4 and operates the properties matrix.

Preliminary step. Create the properties matrix of dimensions $N \times M$ and fill it with zeros and ones according to the following rule: if product i has property j, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. In this matrix, the number of rows will be equal to the number of products and the number of columns to the number of their properties: $N' = N$ and $M' = M$, where N' and M' are auxiliary variables.

Step k. Check whether the properties matrix contains any uncrossed-out rows. If such rows are absent, i.e., $N' = 0$, then the solution is found; terminate the computations. If $N' \neq 0$, then check the existence of a number $1 \le i \le N$ satisfying the relation

$$\sum_{j=1}^{M} a_{ij} = 1, \qquad i = \overline{1, N}.$$

Such products are unique and must be included in the selected set. Cross out row i of the properties matrix and set $N' = N' - 1$; also, cross out the corresponding columns for which $a_{ij} = 1\ \forall j = \overline{1, M'}$.

If there are no unique products, the original matrix will remain unchanged. Calculate the row and column sums for the matrix (either modified or original).

Find the row with the largest sum. If there are several such rows, take the row with the smallest number. The product with this number must be included in the set of prototypes, and the rows and columns associated with it must be crossed out. Repeat step k.

Thus, we obtain the solution of the problem in a finite number of iterations.

The number of possible iterations will be surely smaller than the number of products: $k^{\max} < N$. Indeed, each product has several properties, and crossing out the columns corresponding to these properties implicitly reduces the number of products. At the next step, some product with a certain number of properties coinciding with those of the already selected one will not be included in the set of prototypes: the sum of the row corresponding to this product will be non-maximum (or even equal to 0).

Example 1. Consider seven products with twenty properties, i.e., $N = 7$ and $M = 20$. The properties matrix A is given in Table 1.

Table 1

The properties matrix *A* after identifying a unique product

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Σ
I	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
II	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
III	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5
IV	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	6
V	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	6
VI	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	6
VII	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	7
Σ	2	2	3	3	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	

It is necessary to select a certain number of products that have all the twenty properties.

First, we demonstrate the capabilities of an exact algorithm based on the properties of Boolean polynomials [9, 10].

With the reduced problem dimension, the objective function (1) and the system of constraints (3) can be written in the expanded form

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \to \min,$$
$$x_1 + x_2 \ge 1,\ x_1 + x_2 \ge 1,$$
$$x_1 + x_2 + x_3 \ge 1,\ x_1 + x_2 + x_3 \ge 1,$$
$$x_2 + x_3 + x_4 \ge 1,\ x_2 + x_3 + x_4 \ge 1,$$
$$x_3 + x_4 \ge 1,\ x_4 + x_5 \ge 1,$$
$$x_4 + x_5 \ge 1,\ x_4 + x_5 \ge 1,$$
$$x_5 + x_6 \ge 1,\ x_5 + x_6 \ge 1,\ x_5 + x_6 \ge 1,$$
$$x_6 + x_7 \ge 1,\ x_6 + x_7 \ge 1,\ x_6 + x_7 \ge 1,$$
$$x_7 \ge 1,\ x_7 \ge 1,\ x_7 \ge 1,\ x_7 \ge 1.$$

In this case, the expression (4) takes the form

$$\left(x_1 + x_2\right)^2 \times \left(x_1 + x_2 + x_3\right)^2 \times \left(x_2 + x_3 + x_4\right)^2$$
$$\times \left(x_3 + x_4\right) \times \left(x_4 + x_5\right)^3 \times \left(x_5 + x_6\right)^3 \times \left(x_6 + x_7\right)^3 \times x_7^4 \ge 1.$$

The degree of the polynomial is 20, which matches the initial data. For further solution, we choose the Boolean polynomial terms of the maximum possible degree. Note that it is necessary to take only one term of the maximum degree from each cofactor of the Boolean polynomial. In this example, we have $x_7^7,\ x_2^6,\ x_5^6,$ and x_3; another alternative is to choose $x_7^7,\ x_2^6,\ x_5^6,$ and x_4. In other words, the admissible solutions are as follows:

- the first one: products II, III, V, and VII are selected as prototypes;
- the second one: products II, IV, V, and VII are selected as prototypes.

Even this elementary example elucidates the challenges in obtaining a solution from a Boolean polynomial: the procedure of extracting the terms of maximum degree will be difficult to formalize and quite complicated for programming. Therefore, we apply the heuristic algorithm described above.

Preliminary step. Let us build the properties matrix *A* given in Table 1.

Step 1. We check the existence of uncrossed-out rows in the properties matrix. There are such rows. Hence, we identify unique products. This is product VII, which has properties 17–20; it also closes properties 14–16. The corresponding columns and rows in Table 1 have been crossed out to reduce the problem dimension. We calculate the row and column sums; they are presented in Table 2.

Next, we find the row with the largest sum. In this case, there are several such rows: nos. 2, 4, 5, and 6. Selecting the second row, we delete this row and the associated columns from the properties matrix. The result is shown in Table 2.

Step 2. We check the existence of uncrossed-out rows in the properties matrix. There are such rows. Hence, we try to identify unique products. Such products are absent: all column sums differ from 1. Therefore, we find the rows with the largest sum. This is the fifth row with a sum of 6. The corresponding columns and rows in Table 3 have been crossed out to reduce the problem dimension. We calculate the row and column sums; they are presented in Table 3.

Step 3. We check the existence of uncrossed-out rows in the properties matrix. There are such rows. Hence, we try to identify unique products. Such products are absent: all column sums differ from 1. Therefore, we find the rows with the largest sum. There are two such rows: no. 3 and no. 4 with the same sum equal to 1. We include product III in the sample. The results are presented in Table 4.

Table 2

The properties matrix *A* after Step 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	∑
I	1	1	1	1	0	0	0	0	0	0	0	0	0	0
II	1	1	1	1	1	1	0	0	0	0	0	0	0	–
III	0	0	1	1	1	1	1	0	0	0	0	0	0	1
IV	0	0	0	0	1	1	1	1	1	1	0	0	0	4
V	0	0	0	0	0	0	0	1	1	1	1	1	1	6
VI	0	0	0	0	0	0	0	0	0	0	1	1	1	3
∑	–	–	–	–	–	–	2	2	2	2	2	2	2	

Table 3

The properties matrix *A* after Step 2

	7	8	9	10	11	12	13	∑
I	0	0	0	0	0	0	0	0
III	1	0	0	0	0	0	0	1
IV	1	1	1	1	0	0	0	1
V	0	1	1	1	1	1	1	0
VI	0	0	0	0	1	1	1	0
∑	2	–	–	–	–	–	–	

Table 4

The properties matrix *A* after Step 3

	7	∑
I	0	0
III	1	1
IV	1	1
VI	0	0
∑	2	

Step 4. We check the existence of uncrossed-out rows in the properties matrix. There are no such rows, i.e., the solution has been obtained. It is necessary to choose products II, III, V, and VII for the new generation. According to the previous step, another solution is to include product IV instead of product III, i.e., the representatives of the new generation must be products II, IV, V, and VII.

Clearly, both solutions represent all the twenty properties of the original set of products in the model range. ♦

Thus, we have found the solution of the complete set covering problem. The solution of its incomplete counterpart causes certain difficulties: it is unclear what properties will be selected in the end (they are not specified in the original problem statement). In other words, the selected prototypes must possess not the entire set of properties of the product series but only some part of them, and this part has yet to be determined during the solution procedure. This circumstance prevents the application of the sequential approximation method: first, the complete set covering problem is considered; then the cardinality of the set is reduced by one, and the corresponding set covering problem is solved, etc., until reaching the required size of covering. In this case, the question is what properties should be discarded. Of course, the properties inherent in the earliest samples of products can be discarded, following a natural assumption that they have been implemented, in one form or another, in new samples or even have been replaced by advanced similar functions. But this problem turns on the assessment of product properties and their ranking by significance.

3. AN ALGORITHM FOR SOLVING THE PARTIAL SET COVERING PROBLEM

Consider the partial set covering problem for a bipartite graph. Let us define a bipartite graph $G(X, Y, W)$, where X is the set of vertices of the first layer (products), Y is the set of vertices of the second layer (properties), and W is the set of arcs. An arc $(i, j) \in W$ if product i has property j. We denote by $A \subseteq X$ a subset of X by $B(A) \subseteq Y$ a subset of Y containing all vertices adjacent to A. In other words, A covers $B(A)$, or the set of products A possesses in aggregate the properties $B(A)$.

We formulate the partial set covering problem as follows: it is required to find a set A of minimum cardinality such that $|B(A)| \geq m$.

Note that if $m = M$, we obtain the well-known bipartite graph covering problem.

A heuristic (greedy) algorithm yielding an upper bound includes the following steps.

Step 1. Define $i_1 \in X$ of the maximum degree. Remove it and the set $Y_1 \in Y$ of all vertices adjacent to i_1. If $|Y_1| \geq m$, the problem is solved. Otherwise, proceed to the next step.

Step k. Determine i_k of the maximum degree. Remove it and the set $Y_k \in Y$ of all vertices adjacent to i_k. If $\left|\bigcup_{S=1}^{k} Y_s\right| \geq m,$ the problem is solved. Otherwise, proceed to the next step.

Obviously, the problem will be solved in a finite number of steps not exceeding m. The resulting solution gives an upper bound H_{upp}.

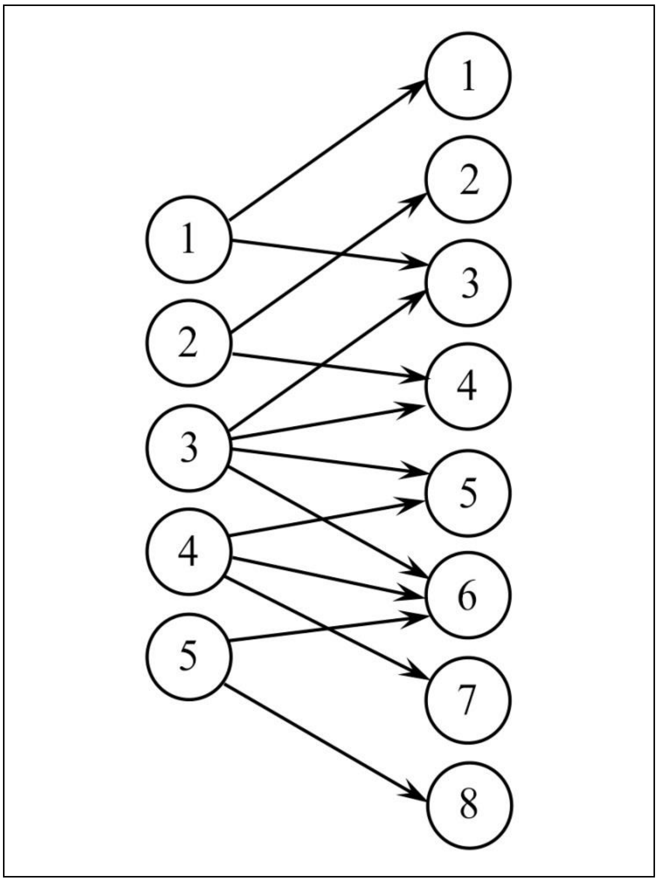

Fig. 1. The graph in Example 2.

Example 2. Consider the graph in Fig. 1. Let $m = 8$ (complete covering).

Step 1. Vertex $3 \in X$ has the maximum degree. We remove it and vertices 3–6 adjacent to it.

Step 2. The residual graph is shown in Fig. 2. Here, the solution is obvious. Taking all vertices 1–5, we obtain $A = (1, 2, 3, 4, 5)$ and $\|A\| = 5$. ♦

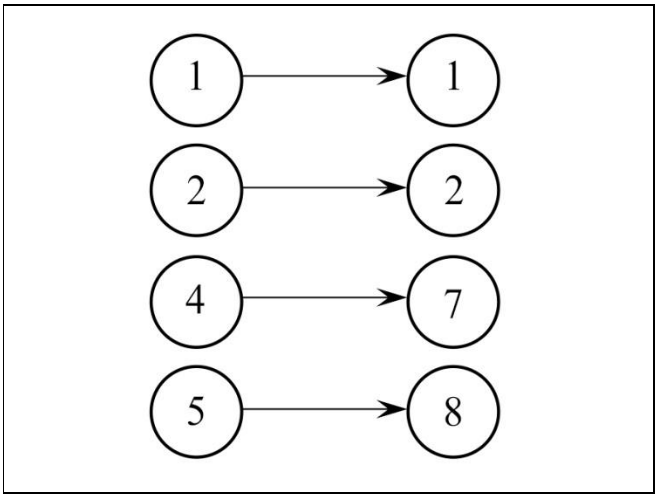

Fig. 2. The graph obtained at Step 2 of the greedy algorithm.

4. THE NETWORK PROGRAMMING METHOD: A LOWER BOUND

To obtain a lower bound, we apply the network programming method [15].

Let us formulate the generalized dual problem. Figure 3 shows the network representation of the original (primal) problem.

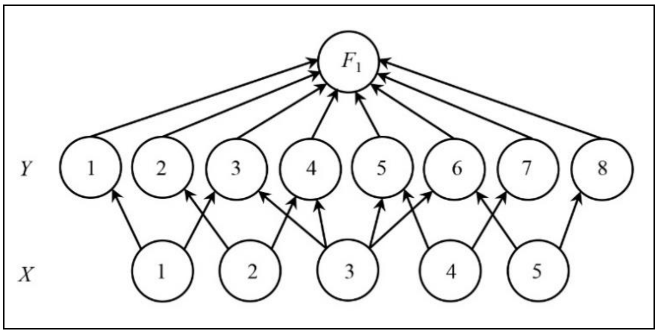

Fig. 3. The network representation of the original problem.

Following the network programming method, the weight of each arc (i, j), where $i \in X$ and $j \in Y$, is given by an arbitrary nonnegative number l_{ij} such that

$$\sum_{j \in P_i} l_{ij} = 1, \quad i = \overline{1, N}.$$

Here, P_i denotes the set of outgoing arcs of vertex $i \in X$.

As a result, we obtain M estimation problems of the form

$$L_i(x) = \min \sum_{i \in Q_i} x_i l_{ij}$$

subject to the constraints $x_i = (0, 1)$ and

$$\sum_{i \in Q_j} x_i \geq 1,$$

where Q_j is the set of incoming arcs from vertex $j \in Y$.

The solutions of these problems are

$$y_j = \min_{i \in Q_j} l_{ij}, \quad j = \overline{1, M}.$$

We rearrange the numbers y_j in ascending order:

$$y_{j1} \leq y_{j2} \leq \ldots y_{jm}.$$

Theorem 1. *The value*

$$H(y) = \sum_{k=1}^{m} y_{jk} \tag{6}$$

gives a lower bound for the original problem.

This theorem is a special case of the general theorem of network programming theory [15].

The generalized dual problem (GDP) is as follows: find the numbers (l_{ij}) maximizing the bound (6).

Example 3. Consider the graph in Fig. 1. The numbers l_{ij} are given in Table 5.

We have

$$y_1 = \frac{3}{4},\ y_2 = \frac{3}{4},\ y_3 = \frac{1}{4},\ y_4 = \frac{1}{4},\ y_5 = \frac{1}{4},$$

$$y_6 = \frac{1}{4},\ y_7 = \frac{3}{4},\ y_8 = \frac{3}{4}, \text{ and } H_{\text{low}} = 4.$$

Note that taking $m = 7$ yields $H(y) = 3\frac{1}{4}$. Anyway, this result is rounded to a lower bound of 4 as well due to the integer values of $F_1(x)$.

This estimate can be used in the branch-and-bound method. ♦

Let us consider an exhaustive search algorithm for solving the problem. It can be applied for small N. Defining a segment $[H_{\text{low}}, H_{\text{upp}} - 1]$ of length $q = H_{\text{upp}} - H_{\text{low}} - 1$, we divide it into two parts: $r = \frac{1}{2}q$ (if q is even) or $r = \frac{1}{2}q$ and $r = \frac{1}{2}q + 1$ (if q is odd). We check all combinations of r elements in a total of N elements. Two cases are possible as follows:

- There exists a combination A such that $B(A) \geq m$. In this case, we divide the segment $[r - 1, H_{\text{low}}]$ into two parts and repeat the procedure.
- No combinations A with $B(A) \geq m$ are available. In this case, we divide the segment $[r, H_{\text{low}} - 1]$ into two parts and repeat the procedure. The optimal solution will be obtained in a finite number of steps. For the graph in Example 1, the resulting bounds are $H_{\text{upp}} = 5$ and $H_{\text{low}} = 4$. Therefore, it is enough to check the combinations of 5 and 4 elements; their number is 5. We find the optimal solution $A = (1, 2, 7, 8)$.

CONCLUSIONS

This paper has considered possible ways to select prototypes for forming a generation of new technological products. The algorithms proposed are based on the set covering problem. They provide solutions in the case of complete and partial set coverage.

The disadvantage of this problem statement is that all the properties possessed by the products are equally important, which, of course, is not true in practice. It would be correct to consider the significance of each property, but it will strongly depend on the target audience: one set of properties is important for product developers and another for ultimate consumers.

Table 5

Initial data for the generalized dual problem

i	1		2		3				4		5	
(i, j)	(1, 1)	(1, 3)	(2, 2)	(2, 4)	(3, 3)	(3, 4)	(3, 5)	(3, 6)	(4, 5)	(4, 7)	(5, 6)	(5, 8)
l_{ij}	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$

REFERENCES

1. Asaturova, Yu.M. and Khvatova, T.Y., Improving Innovative Activity of Enterprises in Conditions of Financial Deficit, *St. Petersburg State Polytechnical University Journal. Economics*, 2019, vol. 12, no. 1, pp. 132–145. (In Russian.)
2. Kurochka, P.N. and Cherednichenko, N.D., Resource Planning Problems in a Construction Project, *Trudy 12-go vserossiiskogo soveshchaniya po problemam upravleniya (VSPU-2014)* (Proceedings of the 12th All-Russian Meeting on Control Problems (AMCP-2014)), Moscow, 2014, pp. 4745–4753. (In Russian.)
3. Novikov, D.A. and Ivashchenko, A.A., *Modeli i metody organizatsionnogo upravleniya innovatsionnym razvitiem firmy* (Models and Methods of Organizational Control of Innovative Development of a Firm), Moscow: KomKniga, 2006. (In Russian.)
4. Belov, M.V., Optimal Control of the Life Cycle of Complex Systems, *Control Sciences*, 2022, no. 1, pp. 15–26.
5. Kurochka, P.N. and Seferov, G.G., A Model of Work-in-Progress Management with an Arbitrary Relationship between Project Activities, *Bulletin of Voronezh State Technical University*, 2011, vol. 7, no. 4, pp. 178–182. (In Russian.)
6. Barkalov, S.A. and Kurochka, P.N., Model for Determining the Term of Execution of Subcontract Works, *Proceedings of 2017 Tenth International Conference "Management of Large-Scale System Development (MLSD),"* Moscow, 2017. DOI: 10.1109/MLSD.2017.8109598.
7. Moncef, E., Stochastic Calculus in a Risk Model with Stochastic Return on Investments, *Stochastics*, 2021, vol. 93, pp. 110–129.
8. Fox, W.P. and Burks, R., Mathematical Programming: Linear, Integer, and Nonlinear Optimization in Military Decision-Making, in *Applications of Operations Research and Management Science for Military Decision Making*, New York: Springer, 2019, pp. 137–191.
9. Knight, F., *Risk, Uncertainty and Profit*, Boston–New York: Houghton Mifflin, 1921.
10. Kaufmann, A., Henry-Labordère, A., *Méthodes et modèles de la recherche opérationnelle. Économie d'entreprise*, Dunod, 1974. (In French.)
11. Christofides, N., *Graph Theory: an Algorithmic Approach*, London: Academic Press, 1975.
12. Selezneva, I.E., Klochkov, V.V., and Egoshin, S.F., Mathematical Model of Intersectoral Coordination of Development Strategies (on the Example of Healthcare and Aircraft Industry, *Large-Scale System Control*, 2022, no. 99, pp. 57–80. (In Russian.)
13. Medvedev, S.N., Greedy and Adaptive Algorithms for Multi-Depot Vehicle Routing with Object Alternation, *Automation and Remote Control*, 2023, vol. 84, no. 3, pp. 341–364.
14. Dranko, O.I., A Financial Prediction Model and Home Investment Scenarios, *Control Sciences*, 2007, no. 1, pp. 37–40. (In Russian.)
15. Burkov, V.N. and Burkova, I.V., *Zadachi dikhotomicheskoi optimizatsii* (Dychotomic Optimization Problems), Moscow: Radio i Svyaz', 2003. (In Russian.)

This paper was recommended for publication by A.A. Lazarev, a member of the Editorial Board.

Received May 22, 2023,
and revised October 29, 2023.
Accepted October 30, 2023.

Author information

Barkalov, Sergei Alekseevich. Dr. Sci. (Eng.), Voronezh State Technical University, Voronezh, Russia
✉ bsa610@yandex.ru
ORCID iD: https://orcid.org/0000-0001-6183-3004

Burkov, Vladimir Nikolaevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ vlab17@bk.ru
ORCID iD: https://orcid.org/0000-0001-6633-3762

Kurochka, Pavel Nikolaevich. Dr. Sci. (Eng.), Voronezh State Technical University, Voronezh, Russia
✉ kpn55@ramler.ru
ORCID iD: https://orcid.org/0000-0003-4945-9552

Serebryakova, Elena Anatol'evna. Cand. Sci. (Econ.), Voronezh State Technical University, Voronezh, Russia
✉ sea-parish@mail.ru
ORCID iD: https://orcid.org/0000-0001-5129-246X

Cite this paper

Barkalov, S.A., Burkov, V.N., Kurochka, P.N., and Serebryakova, E.A., Forming the Generations of New Technological Products as a Set Covering Problem. *Control Sciences* **6,** 18–26 (2023). http://doi.org/10.25728/cs.2023.6.2

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