Pavel Sergeevich Shcherban, Artur Nairievich Karagadyan, Kamilla Rinatovna Dzhalalova*

SEARCHING FOR OIL STORAGE TANK LAYOUT BY SOLVING NP-COMPLETE COMBINATORIAL OPTIMISATION PROBLEM

IZNALAŽENJE RASPOREDA NAFTNIH REZERVOARA REŠAVANJEM NP-POTPUNOG KOMBINATORNOG PROBLEMA OPTIMIZACIJE

Originalni naučni rad / Original scientific paper
Rad primljen / Paper received: 25.08.2023
https://doi.org/10.69644/ivk-2024-03-0398

Adresa autora / Author's address:
Immanuel Kant Baltic Federal University, The Branch Scientific Cluster Inst. of High Technologies, Kaliningrad, Russian Federation
P.S. Shcherban 0000-0001-5106-7852
*email: ursa-maior@yandex.ru

Keywords
- vertical steel tanks
- oil depot
- design and reconstruction
- combinatorial optimisation
- knapsack problem
- bin-packing problem
- Python programming

Abstract

In the process of designing a new tank farm or reconstructing an old one, we often face the task of finding the optimal layout of hydrocarbon storage tanks. It is necessary to comply with both the customer's wishes and the requirements of normative documents regarding restrictions in the location of petroleum product storage facilities, and it must be correlated with the size of the site where the tanks are to be constructed. This results in an optimisation problem that can be solved in various ways. In this research, we shall consider two methods for solving such an NP-complete combinatorial optimisation problem: the knapsack problem and the bin-packing problem. For this purpose, we shall solve the discussed problem using both methods analytically, and then form programme algorithms for the solution based on the Python language. Let us compare both methods in real cases of tank placement and analyse their efficiency.

Ključne reči
- vertikalni čelični rezervoari
- naftna stovarišta
- projektovanje i rekonstrukcija
- kombinatorna optimizacija
- problem ranca
- problem pakovanja u prostoru
- Python programiranje

Izvod

U procesu projektovanja nove farme rezervoara ili pri rekonstrukciji stare, često se suočavamo zahtevima iznalaženja optimalnog rasporeda rezervoara za skladištenje ugljovodonika. Potrebno je istovremeno se usaglasiti sa željama investitora i sa zahtevima normativnih dokumenata koji se odnose na restrikcije u lokacijama postrojenja za skladištenje naftnih proizvoda, a treba i da se uskladi sa veličinom lokacije, predviđene za izgradnju rezervoara. Ovo se svodi na problem optimizacije koji se može rešiti na razne načine. U ovom radu, razmatramo dva metoda za rešavanje takvog NP-potpunog kombinatornog problema optimizacije: problem ranca i problem pakovanja u prostoru. Za tu svrhu, rešavamo opisani problem analitički korišćenjem obe metode, a zatim formiramo programske algoritme rešenja u jeziku Python. Upoređujemo obe metode u realnim slučajevima rasporeda rezervoara i analiziramo njihovu efikasnost.

INTRODUCTION

One of the most important problems for the oil and gas industry is the problem of combinatorial optimisation in engineering and construction. It occurs constantly at various stages of the life cycle of equipment and facilities, in the course of debugging various technological processes, during risk assessment, etc. There is a large number of different mathematical methods that allow solving such problems with a certain accuracy. At the same time, it should be noted that no universal approach has been developed yet, and in each specific case the researcher or designer needs to use one of the existing methods. At the same time, the results of applying different methods may differ significantly. In this regard, applying several methods and further comparison of the obtained results is a common approach to finding the optimal solution.

During the construction and reconstruction of oil depots and oil storage facilities, the optimisation problem arises during the selection of the optimal tank and associated equipment design, during the selection of main tank material and coating material, during the reorganisation of the storage and discharge system, and, in particular, during the creation of a tank layout at a particular site, /1/.

The creation of such a layout scheme usually takes place either at the design stage of the oil depot or during the development of the project for its reconstruction, /2/. At the same time, certain interstate and national regulatory documents impose certain restrictions to the tank layout schemes, which, primarily, are related to the requirements of environmental and fire safety, as well as the requirements for the organisation of access to fuel storage facilities. Let us consider these restrictions one by one and then proceed to the combinatorial optimisation methods that allow us to solve this problem.

REGULATORY REQUIREMENTS FOR SITING VERTICAL STEEL TANKS

Different States and international organisations have their own requirements for the siting of tank farms. As of 2023, there is no unified international ISO standard. There are also no unified environmental and safety requirements for such facilities. As a result, each individual country or supranational entity has its own requirements for tank siting. A number of such requirements are summarized in Table 1.

Table 1. Standards for siting vertical steel tanks at an oil depot.

Country	Normative document	Name
USA	API 620	Design and construction of large, welded, low-pressure storage tanks
	API Std 650:2007	Welded steel tanks for oil storage
	Standard 2610	Design, construction, operation, maintenance and inspection of terminal & tank facilities
EU	EN 14015	Specification for the design and manufacture of site built, vertical, cylindrical, flat-bottomed, above ground, welded, steel tanks for the storage of liquids at ambient temperature and above
Germany	DIN SPEC 26056	Hazardous area at storage tanks and its equipment
India	No PNGRB/Tech/7-T4SPl (1)/2020	Petroleum and natural gas regulatory board regulations
Russia	Code of Practice 155.13130.2014	Warehouses of oil and oil products Fire safety requirements

At the same time, it is obvious that despite the differences in the regulations of tank layout in different countries, this does not prevent the solution of the combinatorial problem. When developing such a solution, it is possible to create a block and open system of constraints that can be adjusted by the user depending on the specific case for which the tank farm layout is performed, /3/.

The restrictions set by the various standards are generally related to the distance between tanks, the grouping of tanks, the number of tank rows, the bunding around the perimeter, the placement of passageways and strapping, /4/.

In some cases (e.g., as in DIN SPEC 26056) it refers to the allocation of risk zones within the tank farm, /5/. It is necessary to note a certain advantage of such an approach in regulating the siting of oil and gas infrastructure facilities, since such a standard not only establishes requirements for siting, but also relates to the theory of risks and risk assessment in case of leakage or fire, /6/.

There are two ways to solve the NP-complete combinatorial optimisation problem of reservoir placement and to process the results based on requirements of the standards. The first way is the initial setting of boundary conditions and solving the combinatorial optimisation problem directly taking into account the constraints of a particular standard. The second way is a free solution of the combinatorial optimisation problem with subsequent imposition of constraints on the obtained solution. The first method allows us to get the result faster (because we do not spend machine time for a complete search of solution variants), the second method allows us to perform a free solution of the combinatorial problem and then use all possible constraints to evaluate the compliance of the solution with the requirements of any standard (i.e., it is more flexible, but it consumes more machine time).

In this study, we apply the second approach because it allows us to vary the boundary conditions based on the requirements of a specific normative document. The main requirements of the Russian Code of Practice 155.13130. 2014 regarding the placement of vertical steel tanks at an oil depot include the minimum allowable distance between tanks of the same group (which is limited either in tank diameters or in metres - from 0.5D to 15 metres), the allowable total nominal capacity of the group (up to 200,000 cubic metres), the arrangement of tanks up to 400 cubic metres into subgroups with a total volume of up to 4,000 cubic metres, distances to the bund (3-6 metres), number of tank rows depending on the maximum volume (from 2 to 4 rows). There are also other requirements (regarding the location of driveways and passageways, bunding).

Thus these provisions provide a kind of boundary conditions for the layout of tank farms. A number of conditions are also given in Table 2.

Table 2. Some requirements for the placement of tanks according to Code of Practice 155.13130.2014.

Tank	Unit nominal capacity of tanks installed in a group (m^3)	Permitted total nominal capacity of the group (m^3)	Minimum distance between tanks located in the same group
With floating roof	50 000 or more	200 000	30 m
	Less than 50 000	120 000	0.5D, but not more than 30 m
With a pontoon	50 000	200 000	30 m
	Less than 50 000	120 000	0.65D, but not more than 30 m
With stationary roof. Product with a flash point of more than 45°C	50 000 or less	120 000	0.75D, but not more than 30 m
Same. Product with a flash point of more than 45°C	50 000 or less	80 000	0.75D, but not more than 30 m

For small tanks with capacities up to a few thousand litres, the protection zone can be in the range of 1 to 3 metres around the tank. For larger tanks with a capacity of several thousand to several tens of thousands of litres, the size of the protection zone may be 5 to 10 metres or more.

Thus, we have to take into account a tank and an alienated area around the tank (basically a ring). A schematic representation of this is shown in Fig. 1.

Finding an optimal solution for the placement of a group of tanks is a classical optimisation problem. This problem can be solved both in three-dimensional space (the height of tanks can be varied) and in two-dimensional space, and we can talk about the productive area occupied by tanks in relation to the total area, /7/. We will consider this problem as a two-dimensional problem to simplify the calculations, taking into account the limited size of tanks in terms of height.

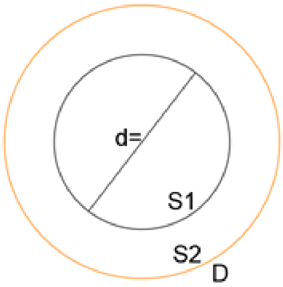

Figure 1. Basic parameters of the tank base and surrounding area. d-diameter of the tank; D-diameter of the protection zone around it (according to Code of Practice 155.13130.2014); S1-productive area occupied by the tank; S2-total area occupied by the tank, taking into account the protection zone.

As a result, in a two-dimensional problem for tanks of different volumes, the only thing that changes are the corresponding areas (Fig. 2). Thus, a number of requirements of the normative document (in this case, Code of Practice 155. 13130.2014, sections 7.2, 7.3, 7.5, 7.6.) and the size of the built-up site become restrictions for the generated linear programming problem. The placement of tanks is carried out without overlapping and going beyond the boundaries of the site. By using a set of similar tanks, with a known system of constraints imposed by the standards and with known site parameters we need to find an optimal solution.

As a basic example, we will consider vertical steel tanks without pontoon or floating roof which are used for storage of petroleum products with flash point less than 45 °C. In this case, for the smallest tanks in the range, we can calculate the following productive and protected areas (Fig. 2).

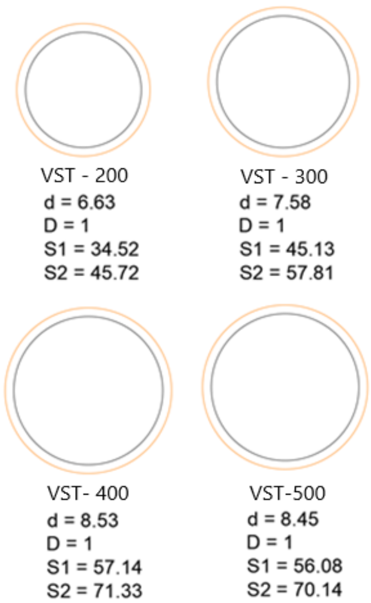

Figure 2. Parameters for small tank bases and surrounding protection zone.

In addition, we note that one of the important conditions is that the site can be filled only with whole tanks (half or a quarter of the tank cannot be installed, but a smaller tank can be installed). Thus, we are faced with the problem of integer linear programming, /8/.

It is not reasonable to carry out the placement by a complete search (given the large number of options and possible complex configuration of the site). Consequently, it is necessary to use a certain mathematical method of searching for the optimal layout of objects.

Mathematical problems of NP-complete combinatorial optimisation provide a similar mechanism. In order to find the optimal placement of tanks, let us consider the varieties of NP-complete combinatorial optimisation methods, select the most effective of them, which will be best suited to the problem under consideration. Use these methods for analytical and software calculations and compare the obtained results.

COMBINATORIAL OPTIMISATION OF OIL STORAGE TANK PLACEMENT THROUGH SOLVING 0-1 KNAPSACK AND BIN-PACKING PROBLEMS

In our case, the most convenient methods for solving the NP-complete combinatorial optimisation problem are the 0-1 *knapsack problem* and the *bin-packing problem* (BPP) method. These methods are chosen as a mathematical model of our problem in connection with their most accurate correspondence to the meaning of the problem being solved and the properties of the considered objects - reservoirs, /9/. In particular, they imply a limited space for placing objects and endowing them with a ‘cost’, which serves as a quantitative indicator for evaluating the effectiveness of the results obtained. Table 3, /10/.

Table 3. Types of NP-complete combinatorial optimisation problems.

Problem	Description
The knapsack problem with an unlimited number of items	In this problem, we need to select some items in order to maximize their value. The number of each item is not limited. This means that we can choose each item multiple times.
Multidimensional knapsack problem	In this problem, there are several knapsacks, each with its own maximum capacity and set of items. We need to choose the items that can be packed into these backpacks in order to maximize their total value.
Bin-packing problem	In this problem, there are a finite number of bins of a fixed size and a set of items, each having a different size. We need to distribute the items among the bins to minimize the number of bins used.
Partitioning problem	In this problem, we need to divide a given set of numbers into two groups so that the sum of the numbers in each group is approximately equal.
0-1 knapsack	The objective is to maximize the total value of the items packed in the knapsack, assuming a limited capacity of the knapsack.
Knapsack problem	The problem of selecting a particular set of items having a limited knapsack capacity is also related to the problem of packing items into bins.
‘Bin-packing with uncertainty’	The problem of packing items into bins of uncertain sizes, weights and shapes, using methods of probability theory and statistics.

In this case, the ‘value’ or ‘cost’ of an object, i.e., of a tank is its capacity, and the limited space for placement is the territory of the oil depot used for tank installation. Thus, the problem we are studying is adequately modelled by the BPP and 0-1 knapsack methods.

In both the 0-1 knapsack problem and BPP, the solution can only be found by searching all possible variants, which is infeasible in most cases, /11/. However, there are various efficient algorithms for these problems that allow us to find an approximate optimal solution.

The analytical solution of the problem using the 0-1 knapsack problem means that we need to fill the territory of area A with objects of type j ($j = 1, n$), i.e., to fill it with tanks of area a_j while maximizing the value of the territory (in the considered problem, maximizing the volume of stored hydrocarbon, which directly depends on the usable area occupied by the tank).

There is an area of the tank of the j-th type: a_j. The value of the tank of type j, which are taken in quantity x_j, can be determined by the function $f_j(x_j)$, /11/. In this case, the items (tanks) cannot be divided into parts.

As a result, we obtain the following mathematical model:

$$f(x) = \sum_{j=1}^{n} f_j(x_j) \to \max ,$$
$$\sum_{j=1}^{n} a_j x_j \le A , \qquad (1)$$
$$x_j \ge 0, \quad x_j \text{ - integer} .$$

For the reason that this problem is an integer programming problem, we can assume that all parameters of the problem have integer values, /12/.

Let us consider this problem as a multistage process. We define a connection with stage k of the problem of placing tanks of the k-th kind in the area. Before the k-th stage occurs, the preceding stages are assumed to have passed as k–1 stages. Since we do not know the area of the oil depot, which will contain tanks of 1, 2, ..., k-th types, in this case, at the k-th stage we carry out the solution of the optimal stacking problem replacing any admissible part of the area, which is described by the parameter $a = 0, 1, 2, ..., A$.

We solve a set of problems that depend on the parameter $a = 0, 1, 2, ..., A$:

$$f(x) = \sum_{j=1}^{k} f_j(x_j) \to \max ,$$
$$\sum_{j=1}^{k} a_j x_j \le A , \qquad (2)$$
$$x_i \ge 0, \quad x_i \text{ - integer} .$$

We denote the optimal value of the problem criterion as $R_k(a)$. In this case $R_k(a)$ defines the maximum value of a part of the area a of the oil depot, where tanks of 1, 2, ..., k-th types are located.

While using the dynamic programming approach, we need to express $R_k(a)$ through the value of $R_{k-1}(a)$, calculated in the previous step, /13/.

For this purpose, we can derive the following recurrence equation:

$$R_k(a) = \max_{\sum_{j=1}^{k} a_j x_j \le a} \sum_{j=1}^{k} f_j(x_j) = \max_{a_k x_k \le a} \Big[f_k(x_k) +$$
$$+ \max_{\sum_{j=1}^{k} a_j x_j \le a - a_k x_k} \sum_{j=1}^{k-1} f_j(x_j) \Big] = \max_{a_k x_k \le a} \Big[f_k(x_k) +$$
$$+ R_{k-1}(a - a_k x_k) \Big] . \qquad (3)$$

As a result, for the 0-1 knapsack problem, the following recurrent dynamic programming equations can be obtained:

$$R_k(a) = \max_{a_k x_k \le a} \left[f_k(x_k) + R_{k-1}(a - a_k x_k) \right], \text{ for } k = 2 \ldots n,$$
$$R_1(a) = \max_{a_k x_k \le a} f_1(x_1) . \qquad (4)$$

It should be noted that in all the Eqs.(1-4) it is assumed that $x_i \ge 0$, x_i - integer. As a result, at the k-th step, we can determine $R_k(a)$ for all values using Eqs.(1) and (4). In this case, we do it by selecting the control solution $x_k = x_k(a)$ and using the values of $R_{k-1}(a)$.

This means that the decision $x_k(a)$ taken at the k-th stage depends solely on the state parameter a, /14/. Actually, returning to the initially set conditions, we can see that the parameter a is the useful area alienated for the tank location, i.e., the maximal a is determined.

The analytical solution to the problem using the BPP method assumes that items of different sizes must be packed in a finite number of containers or bins, each with a fixed given capacity, in such a way as to minimize the number of bins used, /15/.

In this case, we have objects of different sizes, namely vertical cylindrical tanks with different useful area and capacity, as well as fixed specified capacity for filling, namely the existing site for construction or reconstruction.

In general, the BPP problem model is as follows:

$$K = \sum_{j=1}^{n} y_j , \qquad (5)$$
$$\sum_{i=1}^{n} x_i s_i \le y_1, \quad j = 1, 2, \ldots, m ,$$
$$x_i \in \{0,1\}, \quad i = 1, 2, \ldots, n , \quad y_i \ge 0, \quad j = 1, 2, \ldots, m$$

where: n - number of items; m - number of bins (sites); s_i - size of 'i'-th item; x_i - binary variable taking the value 1 if 'i'-th item is packed in a bin (installed on a site) and 0 otherwise; y_i - continuous variable representing the remaining capacity of 'j'-th bin (site);

The goal is to minimize the number of sites required to place all objects, assuming a limited capacity of each site /16/. Given that the problem being solved primarily considers the optimal placement of cylindrical objects on the site (circles on the plane), we can say that this problem is solved by a subsection of the BPP method, namely 2D-CPP (2D Circular Packing Problem), /17/.

For this type of problem it is possible to obtain a more detailed model with additional constraints concerning the filling of the site without intersection of circles (tank bases).

Let us consider the filling of a square site with tanks, where the proposed approach can be modified for a rectangular or complex site (by dividing it into rectangles or squares).

We assume that there are n sites with the same side lengths L. Let $C = \{C_i | 1 \le i \le n\}$ be a set of circular bases of tanks with radii $r_1, r_2, \ldots, r_n$ ($r_i \ge r_{i+1}$). The mandatory condition is that the size of each tank is smaller than the allocated site: $L \ge 2\max\{r_i | 1 \le i \le n\}$. It is necessary to set the number of sites K ($K \le n$) and the partitioning into K - areas $S_1 \cup S_2 \cup \ldots S_k$ so that all circular bases in each set $S_k (1 \le i \le n)$ can be placed on the site B_k, without one overlapping the other, /18/. Let X_{ik} be an indicator of whether the i-th tank is placed on B_k. Y_{ik} will be an indicator whether the site B_k is fully used, /19/. Let the centre of each k-th site be located at the point $(0,0,k)$ in the three-dimensional Cartesian coordinate system, and (x_i, y_i, k) be the coordinates of the centre of the circular base C_i $(1 \le i \le n)$, if $X_{ik} = 1$.

As a result, we obtain the following model:

$$\min K = \sum_{k=1}^{n} Y_k \,, \tag{6}$$

$$\sum_{k=1}^{n} X_{ik} = 1, \quad i \in \{1,\ldots,n\} \,, \tag{7}$$

$$Y_k = \begin{cases} 1 \text{ if } \sum_{k=1}^{n} X_{ik} > 0 & k \in \{1,\ldots,n\} \\ \text{otherwise } 0 & \end{cases} \tag{8}$$

$$\max(|x_i| + r_i, |y_i| + r_i) \le 0.5L, \quad i \in \{1,\ldots,n\} \,, \tag{9}$$

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \ge (r_i + r_j) \quad \text{for} \quad X_{ik} = X_{jk} = 1,$$

$$i, j = \in \{1,\ldots,n\}, \quad i \ne j, \quad k \in \{1,\ldots,n\} \,, \tag{10}$$

$$X_{ik} \in \{0,1\}, \quad i \in \{1,\ldots,n\}, \quad k \in \{1,\ldots,n\} \,. \tag{11}$$

The simplest way to solve it is to use the Greedy Algorithm based on Corner Occupying Action (GACOA), which is based on installing the largest cylinder base in the corner of the site meeting the condition $L \ge 2\max\{r_i | 1 \le i \le n\}$, then the installation process continues sequentially until the maximum filling of B_k; at the same time a cylinder base should not overlap the others, /20/.

In general, it is necessary to note that the solution of the obtained NP-complete combinatorial optimisation problem both with the help of 0-1 knapsack problem and with the help of BPP method, namely 2D-CPP, may not be always optimal, especially if we deal with complex configuration of sites (different from square or rectangle), but this disadvantage is often levelled at the next stage, when the obtained solutions for the placement of tanks are subject to the constraints of standards and rules.

Next, let us proceed to the programme solution of the problem of tank placement on the site using both 0-1 knapsack and BPP problems.

DEVELOPMENT OF SOFTWARE ALGORITHMS FOR SOLVING THE PROBLEM OF TANK PLACEMENT OPTIMIZATION IN PYTHON

The described analytical approach to solving the 0-1 knapsack problem will be formalised in the form of a corresponding algorithm (Fig. 3) in Python. The goal of the algorithm is to maximise the total cost of objects placed in the knapsack (on the available oil depot site), subject to the limited capacity of the knapsack (site).

This algorithm finds the first variant of optimal placement of vertical steel tanks on the territory of the oil depot. Let us present a part of the programme code of the 0-1 knapsack algorithm in Python:

```
for i in range(1, N + 1):
  for j in range(0, W + 1):
    S[i][j] = S[i - 1][j]
if m[i] <= j:
  S[i][j] = max(S[i][j], S[i - 1][j - m[i]] + c[i])
```

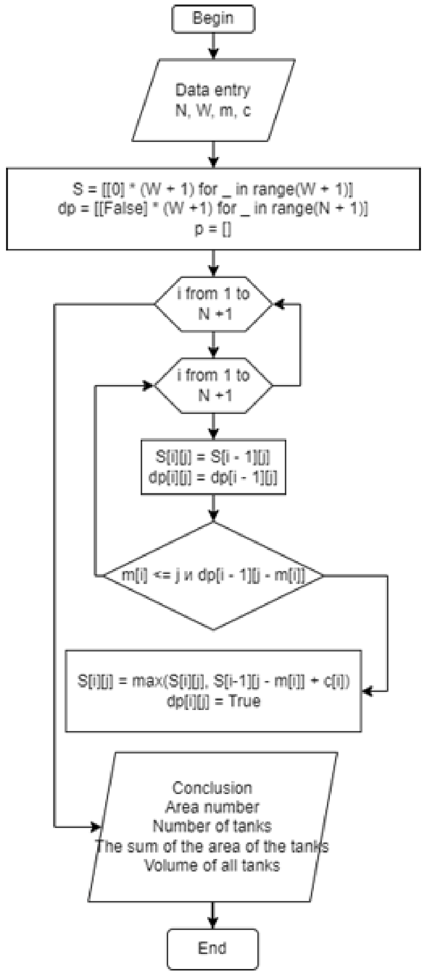

Figure 3. Algorithm of solving the problem of tank placement on the site using the 0-1 knapsack problem in Python.

The second programme algorithm is the bin-packing problem. The idea is to minimise the number of ‘bins’ required to pack all items, provided that the capacity of each bin is limited (i.e., to minimise the amount of unproductively used area of the oil depot site being studied for reconstruction or new construction), /21/. As a result, we developed the following algorithm in Python (Fig. 4) to solve the problem of optimising the placement of tanks using the bin-packing problem (based on the previously given analytical method of solution), /22/.

Here is a part of the programme code of the bin-packing problem algorithm in Python:

```
x = {}
for i in data['items']:
  for j in data['bins']:
    x[(i, j)] = solver.IntVar(0, 1, 'x_%i_%i' % (i, j))
y = {}
for j in data['bins']:
  y[j] = solver.IntVar(0, 1, 'y[%i]' % j)
for i in data['items']:
  solver.Add(sum(x[i, j] for j in data['bins']) = 1)
for j in data['bins']:
  solver.Add(sum(x[(i,j)] * data['weights'][i]
for i in data['items']) <= y[j] * data['bin_capacity'])
```

Thus, we obtain the second solution for the optimal placement of vertical steel tanks on the territory of the oil depot.

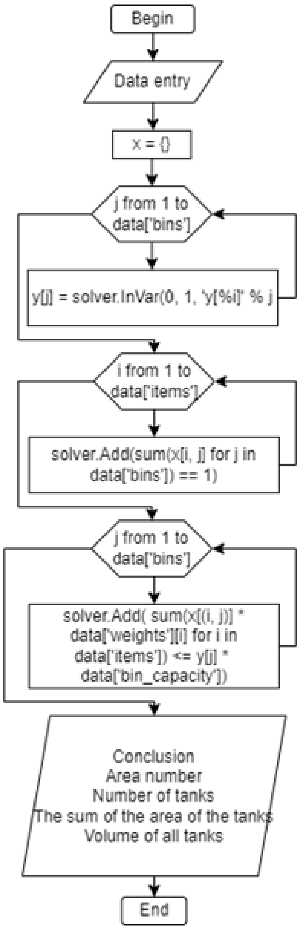

Figure 4. Algorithm for solving the problem of tank placement at the site using the bin-packing problem in Python.

In order to compare the efficiency of both algorithms, we need to specify the dimensions of the site where the tanks are to be installed, in addition to the requirements of the standards for tank placement. As an example, we will use the reconstruction sites of the oil depot of Baltiktop LLC, where horizontal tanks will be replaced by vertical ones.

The first site is a simple rectangle with dimensions 32 m × 60 m. The second site has the shape of two adjacent rectangles, having dimensions 25 m × 60 m × 65 m × 40 m. Let's fill it with vertical steel tanks using the 0-1 knapsack problem and the bin-packing problem.

It should be noted that when solving the problem using the 0-1 knapsack problem, it is important when writing the code to correctly define and store the weight or value of items in data structures, since incorrect values can lead to incorrect results of the algorithm. It is also important to properly use indexes and elements of arrays or data structures to avoid errors of overstepping their bounds or accessing values incorrectly. It is necessary to calculate the total weight or cost of the knapsack correctly based on the selected items (matched tanks) to ensure that the solution is correct.

In case of applying the bin-packing problem, it is necessary to remember that an incorrectly formed data structure may lead to long execution time or to poor optimality of the solution. It is also important to avoid incorrect use of loops, conditional statements or violation of the algorithm logic, as it may lead to incorrect results or programme looping, /23/.

Both presented algorithms (Figs. 3 and 4) in general help to avoid these problems, but at the same time require a clear control of the data entry.

It should be noted that the results obtained by both algorithms may be optimal from a mathematical point of view but will not meet the requirements of the standards. As a result, to comply with each of the previously mentioned standards (for example, Code of Practice 155.13130.2014) both algorithms require a common ‘superstructure’ that provides either coordination of the obtained solution and its registration as a final result, or reconfiguration according to the requirements (Code of Practice 155.13130.2014).

Thus, the obtained optimal solution will comply with the regulatory requirements. Let us develop an algorithm for ensuring compliance with the current Russian standard and possible reconfiguration of the solution results obtained by both methods.

CONFIRMATION OF COMPLIANCE OF OBTAINED RESULTS WITH THE REQUIREMENTS OF THE CODE OF PRACTICE 155.13130.2014 AND ITS ANALYSIS

As previously noted in the Code of Practice 155.13130.2014, there are several basic requirements for the tank layout at the site, namely the distances between tanks of different volumes, the number of tanks in a row and their grouping. To meet the requirements of this standard, it is necessary to pass the results obtained in the previous step through the following compliance algorithms, /24/, in terms of distances between tanks and their locations (Fig. 5).

In addition, a check should be made regarding the placement of tanks in rows (Section 7.9. of the normative document in question). This requires an additional algorithm (Fig. 6), which is a logical continuation of the previous algorithm for confirming compliance, /25/.

The output can be either solutions that are consistent with the requirements of the Code of Practice, or reconfiguration will be required.

In the considered case (due to the small size of the sites) reconfiguration is not required; the presented algorithm confirmed the compliance of solutions and the 0-1 knapsack problem and the bin-packing problem to the requirements of Code of Practice 155.13130.2014. We obtained the following configurations of site filling with tanks (Figs. 7 and 8). For the first site, it is obvious that the bin-packing problem method provided a solution with the largest productive area of 1105.95 m^2 > 1087.95 m^2. The total storage volume of the placed tanks is 12,100 m^3 in case A and 12,200 m^3 in case B. The arrangement of tanks on the site is shown in the summary Table 3.

Filling the second site with a more complex configuration shows an even stronger divergence in the obtained optimisation results. The bin-packing problem also provides a better result - 1938,36 m^2 > 1737,18 m^2. The total storage capacity of the installed tanks is 23,000 m^3 in case A and 24,000 m^3 in case B.

Thus, this method is more efficient when searching for an optimal solution to place production facilities on sites with limited space.

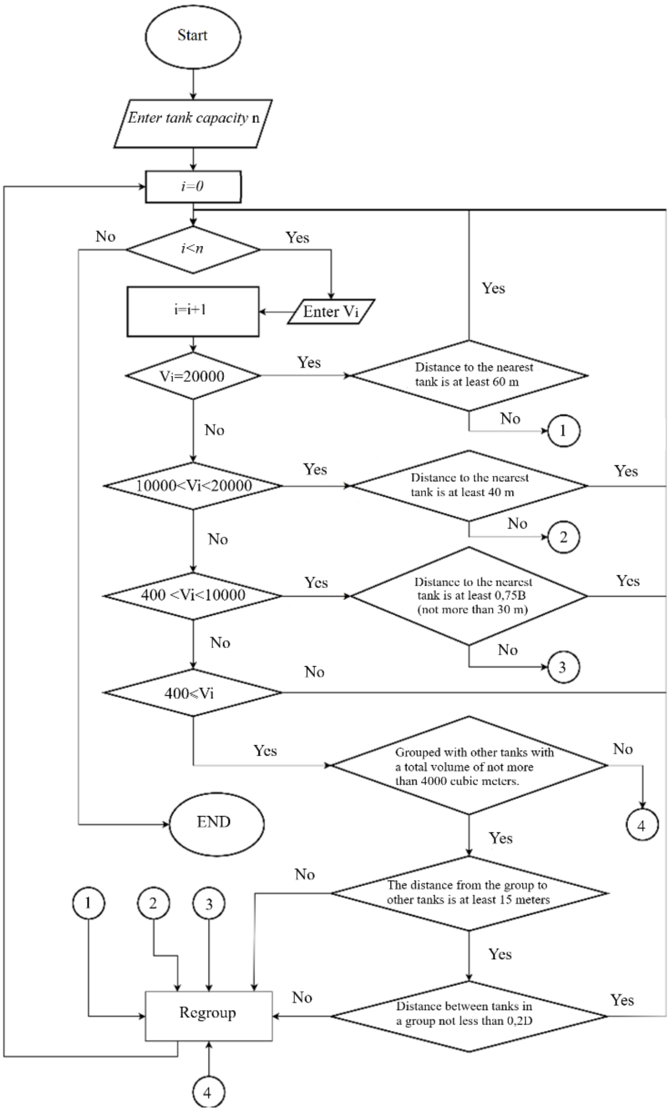

Figure 5. Algorithm for confirming the compliance of the solution results with the requirements of Code of Practice 155.13130.2014 (Sections 7.2, 7.3, 7.5, 7.6) with the possibility of reconfiguration (in Python).

The proposed approach of searching for the optimal solution for the placement of tanks on the territory of the oil depot using the bin-packing problem with the subsequent use of the algorithm of compliance confirmation seems to be the most rational. It should be noted that if other standards are used, the algorithm of compliance confirmation should be restructured.

However, for more complex design problems, it may not be sufficient to implement the solution by the presented approach. For example, the implementation of additional constraints regarding tank piping, infrastructure support, or a more complex site plan for tank placement may require the use of complex optimisation methods with multiple iterations.

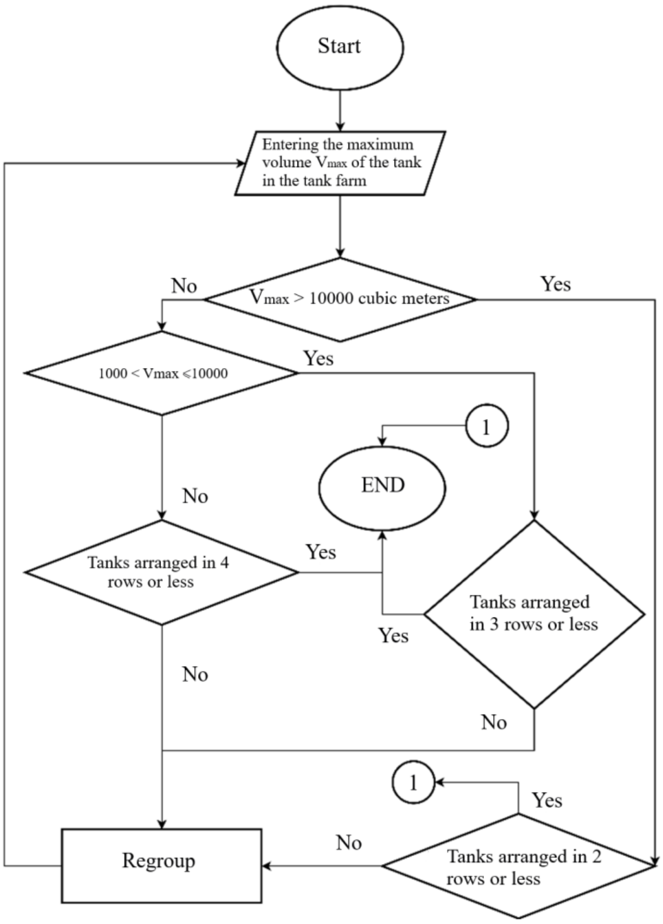

Figure 6. Algorithm for confirming the compliance of the solution results with the requirements of SP 155.13130.2014 (Section 7.9) with the possibility of repackaging (in Phyton).

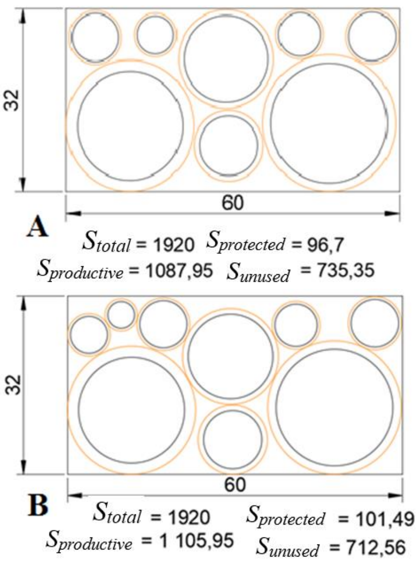

Figure 7. Result of solving the problem of tank placement at site 1 A- by 0-1 knapsack problem; B- by bin-packing problem.

It should also be noted that it took less machine time to solve the bin-packing problem, which also shows the great efficiency of this approach. On the other hand, this result is due to the use of a greedy algorithm, while in the 0-1 knapsack problem a full enumeration is performed, which takes more time.

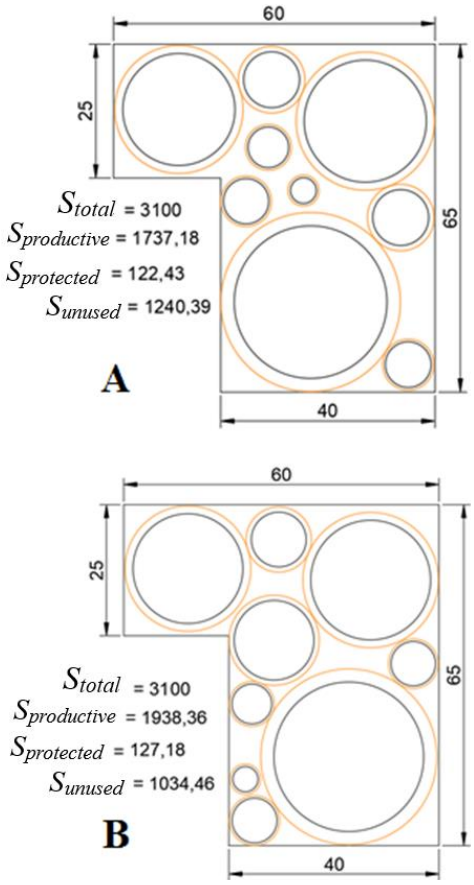

Figure 8. Result of solving the problem of tank placement at site 2 A- by 0-1 knapsack problem; B- by bin-packing problem.

Table 3. Comparison of the performance of 0-1 knapsack problem and Bin-packing problem when searching for the optimal solution of tank placement in an oil depot.

Method	S prod. (m^2)	V total (m^3)	Calc. time (ms)	Tanks placed at the site
1st site				
0-1	1087.95	12 100	1717	VST-200; VST-300; VST-400; VST-500; VST-700; VST-2000; VST-3000; VST-5000
BPP	1105.95	12 200	8	The same + one VST-100
2nd site				
0-1	1737.18	23 000	1736	VST-100; VST-300; VST-400; VST-500; VST-700; VST-1000; VST-5000; VST-5000; VST-10000
BPP	1938,36	24 000	15	The same, but instead of VST-1000, there is VST -2000 placed at the site

CONCLUSION

In the presented research we have developed algorithms to search for the optimal solution for oil storage tank placement using 0-1 knapsack and bin-packing problem methods with the subsequent evaluation of the compliance of the obtained results with the Code of Practice 155.13130.2014.

When designing new or reconstruction of old tank farms and other similar facilities of oil and gas industry, engineers should use existing methods of NP-complete combinatorial optimisation to find the optimal solution.

The authors believe that the use of the bin-packing problem and its variants (2D Circular Packing Problem) is reasonable, since this approach shows the best results of object placement. The site for the placement of tanks is filled more fully. The area of the productive surface occupied by the reservoirs, and consequently, their total volume is larger. Both tasks were solved at first analytically and after by writing programme code in Phyton.

To ensure compliance of the results obtained by the methods of searching for the optimal solution with the current normative requirements, we recommend the use of specialised algorithms given in this study as 'filters.' Algorithm matching is also done by writing code in Phyton. At the same time, the configuration of these algorithms may vary depending on the applied normative requirements.

REFERENCES

1. Stewart, M., Surface Production Operations: Volume 5: Pressure Vessels, Heat Exchangers, and Aboveground Storage Tanks: Design, Construction, Inspection, and Testing, Gulf Professional Publishing, 2021. ISBN 9780128037225
2. Institute, A.P., Design and Construction of Large, Welded, Low-Pressure Storage Tanks, API Standard 620, 10th Ed., 2002.
3. Oyelami, A.T., Olusunle, S.O.O. (2019), *Spherical storage tank development through mathematical modeling of constituent sections*, Math. Model. Eng. Probl. 6(3): 467-473. doi: 10.18280/mmep.060320
4. Roncetti, L. (2011), *Structural and economical optimization of welded steel storage tanks*, In: Proc. of COBEM 2011, 21st Braz. Cong. Of Mech. Eng., Natal, RN, Brazil: 2240-2252.
5. Ricci, F., Scarponi, G.E., Pastor, E., et al. (2021), *Safety distances for storage tanks to prevent fire damage in Wildland-Industrial Interface*, Proc. Safety Environm. Prot. 147: 693-702. doi: 10.1016/j.psep.2021.01.002
6. BS EN 14015:2004, Specification for the design and manufacture of site built, vertical, cylindrical, flat-bottomed, above ground, welded, steel tanks for the storage of liquids at ambient temperature and above.
7. Tarasenko, A.A., Gruchenkova, A.A., Tarasenko, M.A. (2016), *Analysis of differences in the requirements of national regulations and USA standards in the development of the tank bottom differential subsidence*, Oil Industry J. 2016(08): 132-135. (in Russian)
8. Li, C., Conejo, A.J., Liu, P., et al. (2022), *Mixed-integer linear programming models and algorithms for generation and transmission expansion planning of power systems*, Europ. J Operat. Res. 297(3): 1071-1082. doi: 10.1016/j.ejor.2021.06.024
9. Wilbaut, C., Hanafi, S., Coelho, I.M., Lucena, A. (2022), *The knapsack problem and its variants: formulations and solution methods*, In: S. Salhi, J. Boylan (Eds.), The Palgrave Handbook of Operations Research, ch.0, pp.105-151. Springer Cham. doi: 10.1007/978-3-030-96935-6_4
10. Jooken, J., Leyman, P., De Causmaecker, P. (2023), *Features for the 0-1 knapsack problem based on inclusionwise maximal solutions*, Europ. J Operat. Res. 311(1): 36-55. doi: 10.1016/j.ejor.2023.04.023
11. Wilbaut, C., Todosijevic, R., Hanafi, S., Fréville, A. (2023), *Heuristic and exact reduction procedures to solve the discounted 0-1 knapsack problem*, Europ. J Operat. Res. 304(3): 901-911. doi: 10.1016/j.ejor.2022.04.036
12. Dimri, S.C., Kumar, B., Purohit, K.C., Ram, M. (2022), *Application of 0/1-knapsack problem in optimal decision making*, Nonlin. Stud. 29(4): 1215-1222.

13. Zhao, Q., Ren, X. (2023), *Algorithm design and optimization of profile cutting problem based on 0-1 knapsack algorithm*, J Phys.: Conf. Ser. 2483(1): 012027. doi: 10.1088/1742-6596/2483/1/012027
14. Miyamoto, T., Fujiwara, A. (2022), *Robust optimization algorithms for multi-objective knapsack problem*, In: 2022 Tenth Int. Symposium on Computing and Networking Workshops (CANDARW), Himeji, Japan, 2022, pp.430-432. doi: 10.1109/CANDARW57323.2022.00015.
15. Wäscher, G., Haussner, H., Schumann, H. (2007), *An improved typology of cutting and packing problems*, Europ. J Operat. Res. 183(3): 1109-1130. doi: 10.1016/j.ejor.2005.12.047
16. Lewis, R., Song, X., Dowsland, K., Thompson, J. (2011), *An investigation into two bin packing problems with ordering and orientation implications*, Europ. J Operat. Res. 213(1): 52-65. doi: 10.1016/j.ejor.2011.03.016
17. Akeb, H., Hifi, M., Lazure, D. (2013), *A heuristic based algorithm for the 2D circular strip packing problem,* In: S. Fidanova (Ed.), Recent Advances in Computational Optimization. Studies in Computational Intelligence, vol.470, pp.73-92. Springer, Heidelberg. doi: 10.1007/978-3-319-00410-5_5
18. Lodi, A., Martello, S., Monaci, M., Vigo, D. (2014), *Ch. 5: Two-dimensional bin packing problems*. In: V. Th. Paschos (Ed.), Paradigms of Combinatorial Optimization: Problems and New Approaches, 2nd Ed., pp.107-129. doi: 10.1002/9781119005353.ch5
19. Stoyan, Y.G., Yaskov, G.N. (1998), *Mathematical model and solution method of optimization problem of placement of rectangles and circles taking into account special constraints*, Int. Trans. Oper. Res. 5(1): 45-57. doi: 10.1111/j.1475-3995.1998.tb00101.x
20. George, J.A., George, J.M., Lamar, B.W. (1995), *Packing different-sized circles into a rectangular container*, Europ. J Oper. Res. 84(3): 693-712. doi: 10.1016/0377-2217(95)00032-L
21. Castillo, I., Kampas, F.J., Pintér, J.D. (2008), *Solving circle packing problems by global optimization: Numerical results and industrial applications*, Eur. J Oper. Res. 191(3): 786-802. doi: 10.1016/j.ejor.2007.01.054
22. Wang, J., Dowling, A.W. (2022), *Pyomo.DOE: An open-source package for model-based design of experiments in Python*, AIChE J, 68(12): e17813. doi: 10.1002/aic.17813
23. de Winter, R., Bronkhorst, P., van Stein, B., Bäck, T. (2022), *Constrained multi-objective optimization with a limited budget of function evaluations*, Memetic Comp. 14(2): 151-164. doi: 10.1007/s12293-022-00363-y
24. Tjanaka, B., Fontaine, M. C., Lee, D. H., et al. (2023), *pyribs: A bare-bones Python library for quality diversity optimization*, In: GECCO ‘23, Lisbon, Portugal, arXiv:2303.00191v2. doi: 10.1145/3583131.3590374
25. Corke, P., Robotics, Vision and Control: Fundamental Algorithms in Python, 3rd Ed., Springer Tracts in Advanced Robotics, vol.146, Springer Nature, 2023. ISBN 9783031064692