

Models and methods for One–Dimensional Space Allocation Problem with forbidden zones

G G Zabudsky¹ and N S Veremchuk²

¹Sobolev Institute of Mathematics, Novosibirsk, Russia

²Siberian State Automobile and Highway University, Omsk, Russia

E-mail: ¹zabudsky@ofim.oscsbras.ru, ²n-veremchuk@rambler.ru

Abstract. We consider an extension of the well-known optimization placement problem. The problem is One–Dimensional Space Allocation Problem (ODSAP). The classical formulation of the problem is to place rectangular connected objects on a line with the minimum total cost of connections between them. The extension of the problem is that there are fixed objects (forbidden zones) on the line. The objects are impossible to place in forbidden zones. The placed objects are connected among themselves and with the zones. The configuration of connections between objects is defined by a network. A similar situation arises, for example, when designing the location of technological equipment of petrochemical enterprise. It is necessary to place units of equipment so that the total cost of the pipeline ties was minimal. In this article a review of the models and methods to solve of the classical ODSAP is given. The properties of the problem with the forbidden zones are noted. Models of combinatorial optimization and integer programming for the problem are constructed. Algorithms for finding an approximate solution and branch and bounds are described. Results of computational experiments are reported.

1. Introduction

The problems of finding optimal solutions often need to be solved in practice [5, 16, 17, 22]. As a rule, the problems are difficult to solve. The study of such problems requires the construction of new mathematical models and the development of effective algorithms. This article explores one of the problems of optimal placement of connected objects.

To solve practical problems optimal placement, real objects are often replaced by geometric shapes, such as rectangles [13]. Although that the approximation of the real objects of complex form by rectangular objects has some mistake, it is widely used in practice in aided design of the placement technological equipment of various enterprises. This approach reduces the complexity of calculations in solving the problem. For example, the approximation of objects by rectangles, allows you to apply well developed methods of computational geometry for rectangles.

A number of mathematical models of various types for optimal placement of rectangles is constructed [6, 11, 14, 18, 21]. The most developed models and methods of placing objects between which there are no connections. These are the problems of packing and cutting. Methods of linear and integer programming, heuristic procedures are used to solve them. For example, the problem of packing rectangle objects into a semi–infinite strip on plane with forbidden zones and criterion of minimal length of the strip was considered in [14]. For solving the problem a search algorithm with prohibitions for finding an approximate solution was proposed. In [11] the algorithm of local optimization for placement of connected rectangular objects on



plane with the criterion of minimum summary length of connections between the objects was reported.

An important direction in the placement of rectangles is the so-called regular placement, for example, along the center (red) lines. Methods of integer and dynamic programming were used for solving the problem with two criterion of placement rectangles on parallel lines [21]. One of the known problems of placement connected rectangular objects on the line is the One-Dimensional Space Allocation Problem (ODSAP). The problem is to find a placement of connected rectangular objects on the line with the criterion of minimum total cost of connections [4, 18]. For the solving the problem methods of branch and bounds and dynamic programming are developed.

Currently, it is relevant to study the problems of optimal placement on the line and plane, taking into account the forbidden zones and barriers [2, 3, 9, 15, 23, 24]. Forbidden zone is a region, for example, on the plane, which is forbidden to place objects for any reason. These may be fixed objects natural elements, such as mountains, swamps and others. At of reconstruction of the enterprises, such zones may be, for example, technological equipment and available premises. Barriers are defined as regions in which the placement of objects and carrying out connections across the barriers not allowed. Allowed to lay a communication only through the passages in the barriers.

In the article we describe the results research of the problem of optimal placement of connected rectangular objects on the line in the presence of forbidden zones. The objects are impossible to place in forbidden zones. The objects and the forbidden zones are the rectangles. As since the space for placement have dimension one therefore the objects and forbidden zones are linear segments. The centers of the objects are connected between themselves and with the centers of the zones. The configuration of connections between objects are presented in the form of an undirected network. It is necessary to place the objects outside the forbidden zones so that the total cost of the connections between objects themselves and objects and zones was minimal. It is extension the One-Dimensional Space Allocation Problem (ODSAP). The review of the research of the classical ODSAP is reported. The original continuous problem with forbidden zones for the arbitrary undirected network is reduced to a number of discrete subproblems of smaller dimension [23]. In addition the review of the properties of the extension ODSAP is given. A heuristic algorithm [23] and the branch and bound method [24] for solving of the problem are described. Results of computational experiments on comparison of the efficiency of branch and bound method and a heuristic algorithm are reported. A model of integer programming and IBM ILOG CPLEX package were used in the experiments.

2. Practical application

Today, the problem of optimal placement of objects of different sizes is being intensively studied both in the theoretical aspect and on practice. There are two main areas of practical application to solve such problems. It is the placement of elements of electronic devices and the placement of process equipment. In the first direction the problems of placing elements and the tracking of connections between them are solved [6].

In the design of electronic devices the problem of placement is determined the optimal spatial arrangement of elements on a given surface (switch field). The criteria and restrictions in the problem can be divided into metric and topological. The size of the elements and the distance between them, the size of the switchable zero, the distance between the terminals of the elements, the permissible length of the connections are included in the metric conditions. The number of spatial intersections of connections, the number of interlayer transitions, the proximity of fuel elements or incompatible electromagnetic elements and connections to each other are included in the topology conditions.

To determine the geometry of the connections of the structural elements of electronic devices

there is the problem of tracing. There are three types of connection tracking: wired, printed, and film links. Criteria for the optimal solution of the trace problem can be minimized, for example, the total length of the connections, the number of layers of installation, the number of transitions from layer to layer, interference in the circuit elements.

The focus in the second direction is usually on the placement of equipment and the assessment of links between them. Such problems should be solved not only when designing the placement of technological equipment, but also when carrying out design work, as a rule, at the preliminary design (planning) stage in other industries. This is the location of the enterprises workshops, machine hydraulic elements, facilities for laying oil and gas pipelines, and so on. At the planning stage, a preliminary assessment of the relationships between the placed elements is usually carried out. In particular, this refers to the stage of installation of production, as a result of which the problem of placement of equipment is solved. Here, unlike electronic devices, the tracking phase is not so important, since everything is done in three-dimensional space. A number of factors affect the placement option, for example, the type of store construction, safe operation conditions, transportation conditions for parts (blanks), ease of maintenance and repair, the ability to replace equipment, and compliance with building codes and regulations. Often, the creation of direct driveways, ease of operation and maintenance of equipment requires the "regularity" of placing [21] (along red lines).

Solving problems of the second direction in practical terms is due to the increase in the efficiency of work in various industries, reduction of time costs and reduction of production costs. It is argued that up to half of the production costs in the industry are related to the processing of materials and the efficient placement of equipment [7]. An improvement plan often leads to lower costs for handling and transporting materials, reducing the workload on equipment and speeding up the work process. So, efficient placement of technological equipment is important for the competitiveness of production. To solve these problems in the process of designing a new or upgrading an existing production, mathematical tools and applied software are used.

In the practical application of placement problems it is often necessary to take into account the sizes of objects. Accounting for this factor in an automated solution allows you to choose the best option for placement of equipment, which more adequately reflects the real situation.

3. Models of the ODSAP

3.1. Statement of the problem

The set of numbers of the placed objects and a set of the forbidden zones we will designate through $I = \{1, \dots, n\}$ and $J = \{1, \dots, m\}$ respectively. Every object i and every zone j is the rectangle with dimensions $l_i \times h_i$ and $p_j \times d_j$ respectively, where $i \in I$ and $j \in J$. The centers of objects are connected with each other and with the centers of zones. Connections from the centers are laid to the line on which the placement is performed, and then along this line. It is obvious that the lengths of the vertical components of the connections between any objects i and j and object i and zone j do not depend on the order of the objects placement on the line and are equal $h_i/2 + h_k/2$ and $h_i/2 + d_j/2$ respectively. The non-intersection conditions of objects i and k and object i and zone j mean that the specified distances must be at least $l_i/2 + l_k/2$ and $l_i/2 + p_j/2$. We can include the values of minimum admissible distances the objects between themselves and with zones. Next we assume that the minimum admissible distances between the projections of the object centers and zones are given. The problem is reduced to the placement of points, i.e. projections of centers of rectangles on the line. Denote by r_{ik} , ($r_{ii} = 0$), $i, k \in I$, and t_{ij} , ($t_{ii} = 0$), $i \in I$, $j \in J$, the minimum admissible distances between objects i and k and between object i and zone j respectively $i, k \in I$, $j \in J$. We denote by $R = (r_{ik})$, $T = (t_{ij})$ symmetric matrixes of minimum admissible distances the objects among themselves and with zones respectively.

The configuration of connections between objects is defined by a network $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is a set of nodes and E is a set of undirected arcs. Arc $(v_i, v_k) \in E$, if there is a connection between objects i and k . Let $u_{ik} \geq 0$ ($u_{ik} = u_{ki}$), $w_{ij} \geq 0$ ($w_{ij} = w_{ji}$) are the costs of connections between objects i and k (weight of the arc (v_i, v_k)), and between object i and zone j for $i, k \in I, j \in J$, and $i < k$ respectively. Need to find placement of the objects on the line so that restrictions on the minimum admissible distances between the objects and between the objects and zones were satisfied and the summary cost of connections the objects among themselves and the objects with zones was minimal [23].

Consider a line segment of length LS containing rectilinear forbidden zones with the centers at $b_j, j \in J$. For simplicity, we can assume that the left border of the segment has the coordinate equal to zero. Denote by x_i the coordinate of center of object $i, i \in I$; let $x = (x_1, \dots, x_n)$ is the placement of the objects. It is need to find minimum of the function:

$$F(x) = \sum_{i \in I} \sum_{j \in J} w_{ij} |x_i - b_j| + \sum_{i, k \in I} \sum_{(v_i, v_k) \in E} u_{ik} |x_i - x_k| \rightarrow \min, \tag{1}$$

under constraints

$$|x_i - b_j| \geq t_{ij}, \quad i \in I, j \in J, \tag{2}$$

$$|x_i - x_k| \geq r_{ik}, \quad i, k \in I, i < k, \tag{3}$$

$$\frac{l_i}{2} \leq x_i \leq LS - \frac{l_i}{2}, \quad i \in I. \tag{4}$$

The following conditions for the elements of the matrix R are considered:

- (a) $r_{ik} = \frac{l_i + l_k}{2}, \quad i, k \in I, i \neq k$ (noncrossing conditions);
- (b) $r_{ij} + r_{jk} \geq r_{ik}, \quad i, j, k \in I, i \neq j \neq k$ (metric problem);
- (c) r_{ik} – arbitrary, $i, k \in I$, (non-metric problem).

Specified conditions can be considered for the elements of the matrix T .

3.2. The classical ODSAP

The classical ODSAP is the problem (1), (3) without forbidden zones ($J = \emptyset$) and with conditions (a). The ODSAP is NP-hard when G is an arbitrary unweighted and unoriented network of connections between objects [8, 20].

In this article the focus is on the problem (1)–(4) with conditions (a).

The classical ODSAP problem (1),(3) ($J = \emptyset$) in terms of permutations is formulated as follows. Denote by $\pi = (\pi(1), \dots, \pi(n))$ the permutation of the objects. Let π^{-1} denote the inverse of this permutation: $\pi^{-1}(i)$ is the position of object i in the permutation π . Consider the permutation π and two objects i and j . The distance between i and j with respect to this permutation, assumed to be taken between their centers, is equal to the half-length of object i , plus the lengths of all objects which are between objects i and j in π , plus the half-length of object j :

$$r(i, j, \pi) = l_i/2 + l_j/2 + \sum_{k \in S(i, j, \pi)} l_k,$$

where $S(i, j, \pi)$ is the set of the objects between objects i and j in π .

The problem of finding a permutation π which minimizes the weighted sum of the distances between all pairs of objects is the ODSAP.

$$D(\pi) = \sum_{i \in I} \sum_{j \in I, j \neq i} u_{ij} r(i, j, \pi).$$

The following property is true for the classical ODSAP. There is the symmetry of the solutions to the problem. Let permutation π' be the symmetric for permutation π , i.e.

$$\pi'(t) = \pi(n - t + 1) \quad \text{for all } t = 1, \dots, n.$$

Then $D(\pi') = D(\pi)$. So, we can exchange the right and the left hand sides of the line in our definition. Thus, we could somewhat simplify the problem by only considering, for instance, the permutation in which $\pi^{-1}(1) \leq n/2$.

3.3. Model of mixed-integer linear programming

We describe (1)–(4) as a model of mixed-integer linear programming (MILP) [23]. To search of the minimum of the function (1) without conditions (2) and (3) can be set as linear programming problem with the addition continuous variables: $s_{ij} \geq 0$, $t_{ik} \geq 0$, $i, k \in I$, $j \in J$, $i < k$. To formulate the condition of disjointness of objects with each other and with zones and define their positions relative to each other (to the left or to the right) introduce the Boolean variables: z_{ik}^1 , $i, k \in I$, $i < k$ and z_{ij}^1 , $i \in I$, $j \in J$ ($z_{ik}^2 = 1$, if object i is located to the left of object k , otherwise $z_{ik}^2 = 0$). Subject to designations the MILP model of (1)–(4) is:

$$F(x) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} \cdot s_{ij} + \sum_{i=1}^{n-1} \sum_{k=i+1}^n u_{ik} \cdot t_{ik} \rightarrow \min, \quad (5)$$

$$-s_{ij} \leq x_i - b_j \leq s_{ij}, \quad i \in I, j \in J, \quad (6)$$

$$-t_{ik} \leq x_i - x_k \leq t_{ik}, \quad i, k \in I, i < k, \quad (7)$$

$$\begin{cases} x_i - b_j - (l_i + p_j)/2 + C \cdot z_{ij}^1 \geq 0, & i \in I, j \in J, \\ b_j - x_i - (l_i + p_j)/2 + C \cdot (1 - z_{ij}^1) \geq 0, \end{cases} \quad (8)$$

$$\begin{cases} x_i - x_k - (l_i + l_k)/2 + C \cdot z_{ik}^2 \geq 0, & i, k \in I, i < k, \\ x_k - x_i - (l_i + l_k)/2 + C \cdot (1 - z_{ik}^2) \geq 0, \end{cases} \quad (9)$$

$$s_{ij} \geq 0, \quad i \in I, j \in J, \quad (10)$$

$$t_{ik} \geq 0, \quad i, k \in I, i < k, \quad (11)$$

$$z_{ij}^1 \in \{0, 1\}, \quad i \in I, j \in J, \quad (12)$$

$$z_{ik}^2 \in \{0, 1\}, \quad i, k \in I, i < k, \quad (13)$$

$$\frac{l_i}{2} \leq x_i \leq LS - \frac{l_i}{2}, \quad i \in I. \quad (14)$$

Constant $C > 0$ is required to meet the alternative conditions of the placement of the objects on the line.

4. Methods of solution

4.1. Solving the classical ODSAP

We describe the complexity and methods of the solution of the problem for the following types of network G : chain, rooted tree and bipolar oriented network taking into account the conditions (a), (b) and (c).

Denote by $P(n)$ the set of all permutations of n elements. Arbitrary placement of the objects on the line $x = (x_{i_1}, \dots, x_{i_n})$, $x_{i_k} < x_{i_{k+1}}$, $k \in I$, $k < n$, corresponds to the permutation $\pi(x) = (i_1, \dots, i_n) \in P(n)$. Note that under the condition (c) to obtain an optimal solution of the problem it is not enough to know the corresponding permutation.

If G is an undirected chain and minimum admissible distances satisfy condition (b), then the algorithm for solution of the problem is trivial. Adjacent objects with numbers i and j must be at the distance r_{ij} . In this case the total cost of the connections between objects is

$$\sum_{(v_i, v_j) \in E} u_{ij} r_{ij}.$$

If there is a violation of triangle inequality between minimum admissible distances, then, generally speaking, it is impossible to place all adjacent objects at minimum distances.

Let the numbering of objects be such that the arcs of network G have the form (v_i, v_{i+1}) , $i \in I$, $i < n$. In this case it is sufficient to consider permutation $\pi_0 = (1, \dots, n)$ for finding the optimal solution of the problem.

Therefore, if G is the chain and the elements of the matrix R are arbitrary rational numbers, then the ODSAP is polynomial solvable for conditions (a), (b), (c). Indeed, is it enough to consider the permutation π_0 for which the problem becomes a linear programming problem (LP).

Note that if G is the chain and $u_{ii+1} = Const$, $i \in I$, $i < n$, then it is not necessary to solve the LP problem. It is enough to apply the method of sequentially single placement of the objects on the line. Put, for example, that $x_1 = 0$, and then use the formula $x_s = \max_{1 \leq k \leq s} (x_k + r_{ks})$, $s \geq 2$.

For the fixed order placement of objects on the line and arbitrary network G problem (1)–(4) becomes the problem of LP. Specific permutation of object numbers corresponds to each placement of the objects. Without loss of the generality, for simplicity we consider $\pi = (1, \dots, n)$. Then the problem for a fixed order placement of the objects on the line given by permutation π is

$$\sum_{i \in I} \bar{u}_i x_i \rightarrow \min, \tag{15}$$

under constraints

$$x_j - x_i \geq r_{ij}, \quad i, j \in I, j > i, \tag{16}$$

where the values $\bar{u}_i = \sum_{k=1}^{i-1} u_{ki} - \sum_{k=i+1}^n u_{ik}$ obtained by the reduction of similar terms at x_i , $i \in I$. The problem (15)–(16) is LP problem. Thus, to solve (1), (3), (4) it is sufficient to solve LP problems corresponding to all possible order of objects and choose the best solution. Since in general the number of permutations of objects is $n!$, it is necessary to reduce the number of permutations considered in solving the problem. For example, it is possible to use local algorithms (rearrange objects within a neighborhood).

Note that the dual for problem (15)–(16) is the problem of finding the optimal flow in some network. In this case, there are no restrictions on the throughput of arcs. The problem dual to (15)–(16) is:

$$W(y) = \sum_{i \in I, i < n} \sum_{j \in I, j > i} r_{ij} y_{ij} \rightarrow \max,$$

under constraints

$$\sum_{j \in I, j < i} y_{ij} - \sum_{j \in I, j > i} y_{ij} = \sum_{j \in I, j < i} u_{ij} - \sum_{j \in I, j > i} u_{ij}, \quad i \in I,$$

$$y_{ij} \geq 0, \quad i, j \in I, j > i.$$

We present an LP problem equivalent to (15)–(16) in which there are no object coordinates. Suppose, as before, $\pi = (1, \dots, n)$. We introduce $n - 1$ continuous variables $p = (p_{ii+1})$, $i \in I$,

$i < n$ defining distances between adjacent objects arranged according to the permutation π . The problem equivalent to (15)–(16) is:

$$\sum_{i \in I, i < n} \left(\sum_{j \in I, j \leq i} \sum_{k \in I, k \geq i+1} u_{jk} \right) p_{ii+1} \rightarrow \min,$$

under constraints

$$\sum_{i=k}^l p_{ii+1} \geq r_{kl+1}, \quad k, l \in I, \quad k \leq n-2, \quad k+1 \leq l \leq n-1,$$

$$p_{ii+1} \geq r_{ii+1}, \quad i \in I, \quad i < n-1.$$

The classical ODSAP was formulated in terms of integer programming [10]. Such formulation has $n(n-1)/2$ binary variables and $3[n(n-1)/2]$ constraints. This integer programming approach is possibility for getting optimal solutions only for small problems. The approach using integer programming models and methods is not efficient because does not take into account the configuration of the connections between objects.

The ODSAP was formulated as a generalization of the Linear and Quadratic Assignment Problems (QAP) [12]. An interesting placement problem closely related to the ODSAP is the one-dimensional version of the QAP of the Koopmans–Beckman type: given n points P_1, \dots, P_n with coordinates a_1, \dots, a_n on a line, and given n objects and a matrix of connections between objects, find a one-to-one assignment of the objects to the points with a view to minimize the total weighted distance. This problem is called the Generalized Linear Ordering Problem (GLOP). The generalization is that conditions $|a_i - a_j| \neq 1$ are satisfied in this problem. Dynamic programming was used to solve the problem.

To solve the classical ODSAP, polynomial algorithms for the cases when G is a rooted tree and a parallel-sequential network taking into account the conditions (a) were proposed in [1, 20]. If the minimum admissible distances satisfy the constraints (b) then for the specified networks the problem becomes NP-hard [19]. Thus, the transition from the non-intersection conditions of objects to the minimum allowable distances satisfying the triangle inequality translates the problem moves from the polynomial solvable class of the problems to the NP-hard class of the problems.

4.2. Solving the ODSAP with forbidden zones

Next we consider NP-hard problem (1)–(4) with conditions (a) of non-intersecting objects with each other and with zones for the case when G is an undirected network. The area of acceptable solutions to the problem (1)–(4) consists of the disjoint segments (blocks). Consider an arbitrary block B_k .

In [23] the algorithm for finding an approximate solution of the problem consists of two stages is proposed. At the first stage, we find the feasible partition of objects into blocks, and at the second stage, the objects in the blocks are rearranged in order to minimize the total cost of connections. Pay attention to some problem properties.

Considering the possible location, the *remainder* in the B_k block is a non-zero length segment between two adjacent elements (objects, zones) which do not have a common border or between the boundary of B_k and the adjacent block. Two elements (objects, zones, remainders) are called *glued* if they have a common border.

Let $x = (x_1, \dots, x_n)$ be a valid solution to problem (1)–(4); $I_k(x)$ is the set of numbers of objects in the B_k block; $\Delta_k(x)$ is the set of remainders in B_k ; n_k is the capacity of the set $I_k(x)$. Note, that x can be represented as $x = (x^1, \dots, x^r)$, where x^k are the coordinates of the placed objects in B_k . In [23] it was proved that for a possible solution of the x problem we can find

another possible solution of x' , such what $|\Delta_k(x')| \leq 1, k = 1, \dots, r$ and $F(x') \leq F(x)$. So, in each B_k , it suffices to consider no more than one remainder. Thus, the original continuous problem is reduced to a discrete problem.

The coordinates of the left and of the right boundaries of B_k (imaginary objects O_L and O_R) denote by BL and BR ; J_{BL} and J_{BR} are the sets of zones to the left and to the right of B_k ; I_{BL} and I_{BR} are the sets of objects to the left and to the right of the block B_k , respectively. So, for a fixed partition objects in blocks, the objective function $F(x)$ can be represented as

$$F(x) = \sum_{k=1}^r F_k(x^k) + Const,$$

where

$$F_k(x^k) = \sum_{s \in I_k(x)} \sum_{t \in I_k(x), t > s} u_{st} |x_s - x_t| + \sum_{s \in I_k(x)} |x_s - BL| \left(\sum_{j \in J_{BL}} w_{sj} + \sum_{i \in I_{BL}} u_{si} \right) + \sum_{t \in I_k(x)} |x_t - BR| \left(\sum_{j \in J_{BR}} w_{tj} + \sum_{i \in I_{BR}} u_{ti} \right).$$

The sum of the costs for connections between objects in B_k is the first component of $F_k(x^k)$, the total cost of connections between objects from B_k and BL , and B_k and BR respectively are the second and the third components of $F_k(x^k)$.

An admissible solution to problem (1)–(4) will be called the local minimum of the problem if $F(x) \leq F(x')$ for $x' : I_k(x) = I_k(x'), k = 1, \dots, r$.

Let us the partition of the objects in the blocks is fixed. Then in every block B_k its possible to consider the subproblem of location $n_k + 2$ objects. In B_k the subproblem contains two imaginary objects O_L and O_R and n_k placed objects. Let us denote the summary cost of connections between placed objects in B_k and the objects O_L and O_R respectively for every $i \in I_k(x)$ by C_{iL} and C_{iR} and then

$$C_{iL} = \left(\sum_{s \in J_{BL}} w_{is} + \sum_{t \in I_{BL}} u_{it} \right),$$

$$C_{iR} = \left(\sum_{s \in J_{BR}} w_{is} + \sum_{t \in I_{BR}} u_{it} \right).$$

Then the subproblem for B_k is

$$F_k(x^k) = \sum_{s \in I_k(x)} \sum_{t \in I_k(x), t > s} u_{st} |x_s - x_t| + \sum_{s \in I_k(x)} C_{sL} |x_s - BL| + \sum_{t \in I_k(x)} C_{tR} |x_t - BR| \rightarrow \min, \tag{17}$$

$$|x_i - x_k| \geq \frac{l_i + l_k}{2}, \quad i, k \in I_k(x), i < k, \tag{18}$$

$$BL + \frac{l_i}{2} \leq x_i \leq BR - \frac{l_i}{2}, \quad i \in I_k(x). \tag{19}$$

It is necessary to find the coordinates of x^k centers of objects in B_k in order to minimize the cost of connections between the placed objects between themselves and with the objects O_L and O_R .

To find the local optimum of problem (1)–(4) for some fixing the partition of objects into blocks, it suffices to solve r independent subproblems (17)–(19). Thus, the solution of the original continuous problem is reduced to solving discrete subproblems.

4.3. Search for feasible partitions

Here we describe briefly the algorithm of search of the feasible partitions of objects into blocks. It is presented in more detail in [23]. Let $L_1 \geq \dots \geq L_r$ and $l_1 \geq \dots \geq l_n$. To find the initial feasible partition in the given order, we find the initial conditionally feasible block for each object. If there is no such a block then we cancel the membership in the block for the previous objects, and so on. After cancelation, for each of the next objects the search for a conditionally feasible block begins with B_1 . If a feasible partition is not found and all feasible blocks for X_1 are reviewed then the problem has no solution. To find the next feasible partition starting with X_n , we find the next conditionally feasible blocks for the objects [23].

To search for an approximate solution of problem (1)–(4), the stopping criterion may be a runtime, a number of iterations, a finding of an exact solution, or a given accuracy estimate. Note that the number of possible partitions X_i , $i \in I$ into blocks B_1, \dots, B_r is not more than r^n . The remainder in B_k can be considered as an additional located facility X_{n_k+1} , for which $C_{n_k+1} L = C_{n_k+1} R = 0$.

It should be noted that if $u_{st} = 0$, $\forall s, t \in I_k(x)$, $s < t$, then for $\forall k = 1, \dots, r$ local optimum problem (17)–(19) can be found using the polynomial algorithm [23]. If we introduce a partial order of objects in the block, which will be represented as a series–parallel network, then in the block, the problem will be solved by a polynomial algorithm [20].

As a rule, if $\exists s, t \in I_k(x) : u_{st} > 0$, then to solve the subproblem (17)–(19) for a small value of n_k it is possible to use $n_k!$ permutations of objects in the block. Can be used for large values of n_k , for example, the branch and bounds algorithm.

4.4. Minimization of the total costs of connections

Consider the heuristic and branch and bounds algorithms for the minimization of the total costs of connections objects between themselves and with zones.

Algorithm the approximate solution

To minimize the total cost of connections of objects the algorithm A is proposed [23]. Objects in blocks are glued consistently depending on the total cost of connections with objects and zones located left (right) of the block B_k . Let NL_k denote the set of objects that are glued to each other so that the leftmost object is glued to the left border of B_k ; and NR_k , which is glued to the right border of B_k ; and let the objects in B_k be numbered as $1, \dots, n_k$ [23].

Algorithm A

Step 0. $S := I_k(x)$; $NL_k := \emptyset$; $NR_k := \emptyset$.

Step 1. If $(NL_k = \emptyset)$ and $(NR_k = \emptyset)$ then go to Step 2.

If $(t \in NL_k)$ then $C_{iL} := C_{iL} + C_{tL}$, otherwise, $C_{iR} := C_{iR} + C_{tR}$ for all $i \in S$.

Step 2. Define $t : \max_{i \in S} |C_{iL} - C_{iR}| / l_i = |C_{tL} - C_{tR}| / l_t$.

Step 3. If the case of $(C_{tL} \geq C_{tR})$, we put $x_t := BL + \sum_{i \in NL_k} l_i + \frac{l_t}{2}$;
 $NL_k := NL_k \cup \{t\}$;

otherwise, let $x_t := BR - \sum_{i \in NR_k} l_i - \frac{l_t}{2}$; $NR_k := NR_k \cup \{t\}$.

Step 4. $S := S \setminus \{t\}$. If $(S \neq \emptyset)$ then go to Step 1, otherwise the objects are placed in B_k .

Branch and bounds algorithm

The computation of the lower bounds of the objective function and the branching methodology are essential for branch and bounds algorithm (BBA). Here we describe briefly the BBA. It is presented in more detail in [24]. Consider the arbitrary block B_k .

Lower bounds

Denote the sets of numbers of allocated objects in B_k by NO_l , NO_r . For simplicity, we assume that objects in the set NO_l have numbers from 1 to s , and in the set NO_r the number of

objects ranges from $t + 1$ to n_k . Denote by SD the set of admissible locations of objects in B_k ; let $\zeta(SD)$ be the lower bound of the function $F_k(x^k)$ for SD . Then $\zeta(SD)$ can be represented as follows [25]:

$$\zeta(SD) = \zeta_1(SD) + \zeta_2(SD) + \zeta_3(SD).$$

The total cost of connections between the placed objects themselves and with the O_L and O_R objects is the value of $\zeta_1(SD)$. This value is calculated exactly, because the coordinates of all these objects are known. The lower bound of the total cost of connections between unplaced objects with O_L , O_R objects and with placed objects in B_k is the value of $\zeta_2(SD)$. The lower bound of the total cost of connections between unplaced objects themselves is the value of $\zeta_3(SD)$ [25].

Two methods of calculation value $\zeta_2(SD)$ are offered in [25].

First method. The summary cost of connections between the placed objects themselves and with objects O_L and O_R for each $i \in I_k(x) \setminus \{NO_l \cup NO_r\}$ is calculated as follows:

$$SL_i = C_{iL} + \sum_{k \in NF_l} u_{ik}, \quad SR_i = C_{iR} + \sum_{k \in NF_r} u_{ik}.$$

Next location of the unplaced objects in B_k is determined by two variants. The objects are ordered by not increase of the relations SL_i/l_i (SR_i/l_i). The objects consistently are glued together in that order with the most left (right) placed facility in B_k . For simplicity, we assume that the glued unplaced objects have numbers from $s + 1$ to t (from t to $s + 1$). Then

$$\zeta_2(SD) = \zeta_{2L}(SD) + \zeta_{2R}(SD),$$

where $\zeta_{2L}(SD)$ and $\zeta_{2R}(SD)$ are the lower bounds of summary cost of the connections unplaced objects with O_L , O_R respectively and with the placed objects in B_k . The values $\zeta_{2L}(SD)$ and $\zeta_{2R}(SD)$ can be calculated as follows:

$$\zeta_{2L}(SD) = \sum_{q=s+1}^t \left(C_{qL} \sum_{g=1}^{q-1} l_g + \sum_{i=1}^s u_{qi} \sum_{k=i+1}^{q-1} l_k \right),$$

$$\zeta_{2R}(SD) = \sum_{q=s+1}^t \left(C_{qR} \sum_{g=q+1}^{n_k} l_g + \sum_{i=t+1}^{n_k} u_{qi} \sum_{k=q+1}^{i-1} l_k \right).$$

The proof that values $\zeta_{2L}(SD)$ and $\zeta_{2R}(SD)$ are the lower bounds of the summary cost of connections unplaced objects with imaginary objects O_L , O_R and with the placed objects in B_k is similarly to the proof in [18].

Second method. Note that the set $I_k(x) \setminus \{NO_l \cup NO_r\}$ can be represented as the union of non-crossing sets $N_L \cup N_C \cup N_R$, where by N_L , N_C , N_R are designated sets of numbers of objects for which we have the inequalities $SL_i > SR_i$, $SL_i = SR_i$, $SL_i < SR_i$ respectively.

Next objects with numbers from N_L (N_R) are ordered by not increase of the relations $(SL_i - SR_i)/l_i$ ($(SR_i - SL_i)/l_i$). The objects consistently are pasted together in that order with the most left (right) placed facility in B_k . The objects with numbers from N_C are placed between sets of the objects with numbers from N_L and from N_R in any order.

So, for each $i \in I_k(x) \setminus \{NO_l \cup NO_r\}$ the coordinate of the center is determined. Allow $I_k(x) \setminus \{NO_l \cup NO_r\} = \{s + 1, \dots, t\}$. We define the value Z_1 as follows

$$Z_1 = \sum_{q=s+1}^t \left(C_{qL} \sum_{g=1}^{q-1} l_g + \sum_{i=1}^s u_{qi} \sum_{k=i+1}^{q-1} l_k + C_{qR} \sum_{h=q+1}^{n_k} l_h + \sum_{j=t+1}^{n_k} u_{qj} \sum_{v=q+1}^{j-1} l_v \right).$$

The value Z_1 is the lower bound of the summary cost of connections the unplaced objects with objects O_L , O_R and with the placed objects in B_k [25].

So, to calculate of value $\zeta_3(SD)$ it is to consider sets of unplaced objects that are all connected between themselves. Next, for example, by means of viewing of permutations of any three interconnect objects, to find an order of an arrangement of objects in the block with the minimum cost of connections between them.

Branching

We describe the branching procedure [24]. At the first level in the branch–decision tree each of the placing objects with numbers from $I_k(x)$ is glued to the left border of the block B_k . At the second level each of unplaced objects is glued to the right border of the block B_k . So, at the second level one of objects is glued to the left border of B_k , another facility is glued to the right border of B_k . At the subsequent levels unplaced objects are glued to the set of the objects which are glued among themselves, extreme left of which is glued to the left border of B_k [24].

Note that the number of nodes of the branch–decision tree at the first level is equal n_k , at the second level — $n_k(n_k - 1)$, at the third level — $n_k(n_k - 1)(n_k - 2)$, etc. The height of the branch–decision tree is equal to n_k , and the quantity of its trailing nodes is $n_k!$ that corresponds to number of possible permutations of objects in B_k [24].

4.5. Computational experiments

Computing experiment in comparison of the solutions obtained by using the BBA [24] and the heuristic algorithm (A) [23] was carried out. All input data were randomly generated. More than 100 instances of the problem have been solved. The algorithms stops when all local optimal solutions are found or the time to solve the problem was running out. Some results of comparison of the algorithms are presented in Table 1, where notations F_A , F_{BBA} and t_A , t_{BBA} are the objective function values and the average running time (in seconds) of the heuristic algorithm A and the BBA respectively. The average relative error of algorithm A is equal to 3%. The algorithm A finds the solutions faster than the BBA as follows from Table 1. For example, for instance 8 the relative error of A is 0,172% and the running time of A is less that of the BBA more than one order.

In addition, a computing experiment on comparison of the solutions obtained by the BBA and IBM ILOG CPLEX package using the MILP model was made. Three series of the test problems were randomly generated each of which includes 5 problems of the same dimension. For each series, we compared the running times of the BBA and CPLEX package. For the dimensions $|I| = 50$, $|J| = 20$, we could not obtain a solution within time 1000 s using CPLEX. The average running time for the problem by the BBA is 898 s. Note that for the BBA and CPLEX package, the average running time solving of the problem with dimensions $|I| = 5$, $|J| = 3$ is 0,968 s and 0,96 s respectively.

The computational experiment by algorithm A and application of package IBM ILOG CPLEX and the MILP model was carried out. The input data were generated by the random() function. Over 150 problems were tested and the number of forbidden zones was varied from 2 to 10, and the number of objects was varied from 5 to 40. The algorithm A stops when all feasible solutions of the problem are reviewed. Some results of the experiment that are obtained by algorithm A and the package are given in Table 2. Notations F_A and F_{cplex} are the values of the objective functions found by the algorithm and the package. The average relative error of algorithm A is equal 3%.

Some series of the test problems were randomly generated with a uniform distribution each of which includes 15 problems of the same dimensions. We compared the running times of algorithm A and package CPLEX. For the dimension $|I| = 40$, $|J| = 6$, we could not obtain a solution within time 1000 s using package CPLEX. The average running time for solving of the problem by algorithm A is 490 s and error is equal 3%. For algorithm A the average running

Table 1. Comparison of the algorithms A and BBA.

No	n	m	F_A	t_A	F_{BBA}	t_{BBA}	Relative error F , %
1	5	3	12238	1	11892	1	2,91
2	6	3	1844	1	1844	2	0
3	10	3	900,25	2	900,25	8	0
4	10	3	1080	3	1080	7	0
5	10	4	2888	2	2888	4	0
6	10	5	1801	2	1801	3	0
7	10	6	1253	50	1253	56	0
8	15	3	3502	6	3496	136	0,172
9	15	4	4008	5	4008	21	0
10	15	5	4468,5	4	4468,5	10	0
11	15	6	8464	5	8464	40	0
12	15	10	10296	9	10296	20	0
13	20	3	39374,75	12	38810,75	2469	1,453
14	20	10	21533,5	15	21533,5	45	0
15	20	15	19443,5	449	19443,5	724	0
16	30	5	57934	18	57934	231	0
17	30	10	52840	1331	52840	5460	0
18	30	20	128656	26	128656	27	0
19	50	10	181909	86	181909	443	0
20	50	20	430112	269	430112	453	0

time for solving of the problem with dimension $|I| = 5$, $|J| = 5$ is 1.29 s, which is greater than such time of the package equal 0.016 s.

5. Conclusion

In this article we consider the extension of the One-Dimensional Space Allocation Problem (ODSAP) with forbidden zones. The configuration of connections between objects is defined by the oriented acyclic network. The review of the formulations and methods for solving the classical ODSAP are given. A review of the properties of the ODSAP with forbidden zones is reported. The branch and bound method [24] and a heuristic algorithm [23] for solving of the problem were described. Results of computational experiments on comparison of the branch and bound method and a heuristic algorithm were given. In the experiments, the integer programming model of the problem and IBM ILOG CPLEX package were used.

In the capacity of perspectives for further research of this problem, it should be noted the following.

- (i) Modify of the algorithm for constructing of admissible partitions of the objects into blocks, taking into account a partial order between objects.
- (ii) Look for different structures of the network of connections between objects, for which it is possible to develop polynomial algorithms to find of local optimum of the problem.
- (iii) Develop of exact methods, heuristic algorithms and decomposition methods for arbitrary oriented network of connections between objects.

Table 2. Comparison of algorithms A and the package CPLEX.

No	n	m	F_A	F_{CPLEX}	Relative error, %
1	3	3	2113	2113	0,0
2	5	3	1287	1287	0,0
3	5	4	15610	14802	5,5
4	4	3	4751,5	4751,5	0,0
5	4	3	1515	1515	0,0
6	6	3	668	668	0,0
7	6	4	7467	7467	0,0
8	6	5	7841,5	7841,5	0,0
9	5	5	13991	12580	11,2
10	5	5	15459	14519	6,5
11	5	5	14996,5	14996,5	0,0
12	6	5	16277,5	16277,5	0,0
13	6	5	18369	18369	0,0
14	6	5	17646,5	17646,5	0,0
15	8	2	12496,5	11760,5	6,3
16	8	2	27640	26250	5,3
17	7	4	7556	7144	5,8
18	7	4	8172	7657	6,7
19	7	4	7851,5	7851,5	0,0
20	7	4	29638,5	29638,5	0,0

Acknowledgments

The work was supported by the program of fundamental scientific research of the SB RAS No. I.5.1., project No. 0314-2019-0019

References

- [1] Adolphson, D., Hu, T.C.: Optimal linear ordering. *SIAM J. Appl. Math.* **25**(3), 403–423 (1973)
- [2] Bischoff, M., Klamroth, K.: An efficient solution method for Weber problems with barriers based on genetic algorithms. *Eur. J. Oper. Res.* **177**, 22–41 (2007)
- [3] Butt, S.E., Cavalier, T.M.: An efficient algorithm for facility location in the presence of forbidden regions. *Eur. J. Oper. Res.* **90**, 56–70 (1996)
- [4] Chan, A.W., Francis, R.L.: Some layout problems on the line with interdistance constraints costs. *Oper. Res.* **27**(5), 952–971 (1979)
- [5] Dinu, S., Ciucur, V.: Location–allocation models and new solution methodologies in telecommunication networks. *IOP Conf. Ser.: Mater. Sci. Eng.* **145** 082021 (2016) <https://doi.org/10.1088/1757-899X/145/8/082021>
- [6] Erzin, A.I., Cho, J.D.: Concurrent placement and routing in the design of integrated circuits. *Automat. Remote Control* **64**(12), 1988–1999 (2003)
- [7] Foulds, L.R., Hamacher, H.W., Wilson, J.M.: Integer programming approaches to facilities layout models with forbidden areas. *Annals of Operations Research* **81**, 405–417 (1998)
- [8] Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP–Completeness.* (Freeman, San Francisco, (1979); Mir, Moscow, (1982))
- [9] Katz, N., Cooper, L.: Facility location in the presence of forbidden regions, I: Formulation and the case of Euclidean distance with one forbidden circle. *Eur. J. Oper. Res.* **6**, 166–173 (1981)
- [10] Love, R.F., Wong, J.Y.: On solving a One–Dimensional Space Allocation Problem with Integer Programming. *INFOR* **14**(2), 139–143 (1976)

- [11] Panyukov, A.V.: The problem of locating rectangular plants with minimal cost for the connecting network. *Diskret. Anal. Issled. Oper. Ser. 2*, **8**(1), 70–87 (2001)
- [12] Picard, J.C., Queyranne, M.: On the one-dimensional space allocation problem. *Oper. Res.* **29**(2), 371–391 (1981)
- [13] Preparata, F., Shamos, M.: *Computational Geometry: An Introduction*. Springer-Verlag, (1985)
- [14] Rudnev, A.S.: Probabilistic tabu search algorithm for the packing circles and rectangles into the strip. *Diskret. Anal. Issled. Oper.* **16**(4), 61–86 (2009)
- [15] Sarkar A., Batta R., Nagi R.: Placing a finite size facility with a center objective on a rectangular plane with barriers. *European Journal of Operational Research* **179**(3), 1160–1176 (2006)
- [16] Sharifah Shuthairah Syed-Abdullah, Syariza Abdul-Rahman, Aida Mauziah Benjamin, Antoni Wibowo and Ku-Ruhana Ku-Mahamud: Solving Quadratic Assignment Problem with Fixed Assignment (QAPFA) using Branch and Bound Approach. *IOP Conf. Ser.: Mater. Sci. Eng.* **300** 012002 (2018) <https://doi.org/10.1088/1757-899X/300/1/012002>
- [17] Shiloach Y.: A minimum linear arrangement algorithm for undirected trees. *SIAM J. Comput.* **8**(1), 15–32 (1979)
- [18] Simmons, D.M.: One-dimensional space allocation: an ordering algorithm. *Oper. Res.* **17**(5), 812–826 (1969)
- [19] Zabudsky, G.G.: On the complexity of the problem of placement on a line with restrictions on minimum distances. *Russian Math. (Iz. VUZ)* **49**(12), 9–12 (2005)
- [20] Zabudsky, G.G.: On the problem of the linear ordering of vertices of parallel-sequential graphs. *Diskret. Anal. Issled. Oper.* **7**(1), 61–64 (2000)
- [21] Zabudskii, G.G., Amzin, I.V.: Algorithms of compact location for technological equipment on parallel lines (in Russian). *Sib. Zh. Ind. Mat.* **16**(3), 86–94 (2013)
- [22] Zabudskii, G.G., Filimonov, D.V.: An algorithm for minimax location problem on tree with maximal distances. *Proc. of the Second International Workshop "Discrete Optimization Methods in Production and Logistics"* (DOM 2004). –Omsk–Irkutsk, 81–85 (2004)
- [23] Zabudskii, G.G., Veremchuk, N.S.: An algorithm for finding an approximate solution to the Weber problem on a line with forbidden gaps. *J. Appl. Ind. Math.* **10**(1), 136–144 (2016) DOI: 10.1134/S1990478916010154
- [24] Zabudsky, G.G., Veremchuk, N.S.: Branch and Bound Method for the Weber Problem with Rectangular Facilities on Lines in the Presence of Forbidden Gaps. Springer International Publishing AG, part of Springer Nature 2018 A. Eremeev et al. (Eds.): OPTA 2018, CCIS **871**, 29–41 (2018). <https://doi.org/10.1007/97833199380043>
- [25] Zabudsky, G., Veremchuk, N.: Weber problem for rectangles on lines with forbidden gaps. *IEEE Conference 2016 Dynamics of Systems, Mechanisms and Machines* (Omsk, 15-17 Nov. 2016) (2016)