About One–Dimensional Space Allocation Problem with forbidden zones

G G Zabudsky¹, N S Veremchuk²
¹ Sobolev Institute of Mathematics, Pevtsova St., 13, 644043, Omsk, Russia
² Siberian State Automobile and Highway University, Mira pr., 5, 644080, Omsk, Russia
E-mail: zabudsky@ofim.oscsbras.ru, n-veremchuk@rambler.ru

Abstract. This article discusses the One–Dimensional Space Allocation Problem (ODSAP) with the additional conditions. There are fixed objects (forbidden zones) and a partial order of placement objects on the line. You can not place objects in forbidden zones. The structure of connections between objects is determined using a directed graph. Situations of this type arise, for example, when designing the placement of technological equipment of some enterprise. Technological scheme of production determines the order of processing of raw materials. A polynomial–time algorithm for finding a local optimum for a special graph of connections between objects is proposed.

1. Introduction

The problems of optimal placement of various dimensional objects are actively investigated both in theoretical aspect and in practice. Such problems should be solved when performing design work, usually at the preliminary design stage in various industries. This is the placement of the elements of the hydraulic system of the machine, oil and gas facilities and so on. At the stage of planning is usually performed a preliminary valuation of the connections between the placing objects.

The practical need to solve the problems of optimal placement of technological equipment is the requirement to increase the efficiency of various enterprises. In [2] it is argued that up to half of the production costs in the industry are associated with the processing of materials and efficient placement of equipment. Optimal placement of technological equipment of any enterprise 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 solving the problems of optimal placement of various objects in practice usually the objects involved in the placement process are approximated by simple geometric shapes, such as rectangles. This reduces geometric complexity when solving problems, for example, when checking the conditions of mutual intersection of the objects.

To solve problems of optimal placement of rectangles, many different approaches both exact and approximate solution have been proposed [4, 5, 7, 9, 12]. Widely applicable in practice and well designed from a mathematical point of view can be considered the placement of rectangular parts on rectangular blanks, the so–called cutting and packing problems. Exact methods of mathematical programming [4], as well as heuristic methods such as evolutionary methods and
others were used to solve such problems. In [7], the problem of packing non-interconnected rectangles into a semi-infinite strip of minimum length was considered. A search algorithm with prohibitions for finding an approximate solution was proposed. The algorithm of local optimization for placement of connected rectangular objects on the plane was proposed in [5].

Important direction in placement problems is regular placement of the objects, for example, on a line. The tools of integer and dynamic programming [12] were used to construct a set of Pareto-optimal solutions to the problem on parallel lines. A dynamic programming algorithm for placement connected rectangular objects on the line was proposed in [9]. One of the known problems of placement connected objects on the line is the One-Dimensional Space Allocation Problem (ODSAP).

Lately the direction of research is gaining popularity of the problems of placement objects, taking into account the forbidden zones and barriers [8, 13, 14]. Barriers and forbidden zones are defined as regions in which the placement of objects is prohibited. In case of modernization of the plant, the zones and barriers may be, for example, the technological equipment. It is forbidden to make connections inside the barrier, but this is possible in zones.

In this paper, we investigate the problem of optimal placement of different-sized connected objects on the line, taking into account the forbidden zones for the placement of objects and a given partial order of their placement. The objects and the forbidden zones are rectangles (linear segments). The objects are connected with each other and with the centers of the zones. The structure of connections between objects and the partial order of their placement on the line are presented in the form of a directed graph. It is necessary to place objects outside the forbidden zones in such a way that a partial order was executed, and the total cost of connections between objects and zones was minimal. It is known that the original continuous problem for the arbitrary undirected graph can be reduced to a number of discrete subproblems of smaller dimension [13]. A polynomial-time algorithm for finding the local optimum is proposed for the case when the graph of connections between objects is a bipolar directed graph.

2. Problem Formulation

Let \( I = \{1, \ldots, n\} \) and \( J = \{1, \ldots, m\} \) be the number set of placed objects and the number set of the forbidden zones respectively. Every object \( i \) and zone \( j \) is a rectangle with sizes \( l_i \times h_i \) and \( p_j \times d_j \), \( i \in I, \ j \in J \). Without limiting the generality we believe that the centers of the objects are connected with each other and with the centers of the zones. Note, that the lengths of the vertical components of connections between object \( i \) and object \( k \), as well as between object \( i \) and zone \( j \) are equal to \( h_i/2 + h_k/2 \) and \( h_i/2 + d_j/2 \) respectively, and they are independent of the placement of objects. The values \( l_i/2 \) and \( p_j/2 \), \( i \in I, \ j \in J \), can be included in the minimum admissible distances and assume that there are restrictions between the projections of the centers of the objects and the zones. The problem is reduced to the placement of point objects, that is, projections of the geometric centers of the rectangles on the line. Let \( r_{ik} = (r_{ii} = 0), \ i, k \in I \), and \( t_{ij}, (t_{ii} = 0), \ i \in I, \ j \in J \), are the minimum admissible distances between the object \( i \) and the object \( k \) and between the object \( i \) and the zone \( j \), respectively. Denote by \( R = (r_{ik}), T = (t_{ij}) \) the symmetric matrixes of minimum admissible distances between objects and between objects and zones respectively [6].

Since the placed objects are interconnected, the structure of connections between objects is determined using the graph \( G = (V, E) \), where \( V = \{1, \ldots, n\} \) and the edge \((i, k) \in E\), if there is a connection between object \( i \) and object \( k \). Let \( u_{ik} \geq 0 (u_{ik} = u_{ki}), w_{ij} \geq 0 (w_{ij} = w_{ji}) \) be the costs of the connections between the object \( i \) and the object \( k \), as well as between the object \( i \) and the zone \( j \) for \( i, k \in I, \ j \in J \), and \( i < k \), respectively. A partial order can be specified for connected objects on the line, for example, object \( i \) must be to the left of object \( k \), then \((i, k)\) is the arc. It is need to place the objects on the line so that restrictions on the minimum admissible distances between objects and between objects and zones were observed and the total
cost of connections between the objects and between objects and zones was minimal [13].

Given a straight–line segment of length \( LS \) containing the fixed rectilinear forbidden zones with the centers at \( b_j, j \in J \). Let \( x_i \) is the coordinate of center of object \( i, i \in I \); let \( x = (x_1, \ldots, x_n) \) is the placement of objects. It is necessary to minimize the function:

\[
F(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} |x_i - b_j| + \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} u_{ik} |x_i - x_k| \rightarrow \min,
\]

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 restrictions on the elements of the matrix \( R \) are considered:

(a) \( r_{ik} = \frac{l_i + l_k}{2}, \quad i, k \in I, i \neq k \) (non–intersection conditions);

(b) \( r_{ij} + r_{jk} \geq r_{ik}, \quad i, j, k \in I, i \neq j \neq k \) (in the case of metric problem);

(c) \( r_{ik} \) – arbitrary, \( i, k \in I \) (in the case of non–metric problem).

Such restrictions can be considered on the elements of the matrix \( T \).

Problem (1), (3) without forbidden zones \( (J = \emptyset) \) but with conditions (a) is the classical ODSAP. For the case when \( G \) is an arbitrary unweighted and undirected graph of connections between objects the classical ODSAP is NP–hard [11].

To solving the classical ODSAP, polynomial–time algorithms have been developed when \( G \) is a rooted tree [1] and when \( G \) is a parallel–sequential graph [11]. For the cases when the minimum admissible distances satisfy the restrictions (b) (or the more restrictions (c)), then for the indicated graphs the problem becomes NP–hard [10].

The classical ODSAP was formulated in terms of integer linear programming in [3]. According to the results of the numerical experiment, this approach can be used to solve problems of small dimension up to 12 objects. The approach with the use of integer programming models is inefficient, does not take into account the structure of relations between objects.

Problem (1)–(4) with restrictions (a) of non–intersection of objects with each other and with zones, but without partial order between objects, that is, for the case when \( G \) is an undirected graph was considered in [13, 14]. For one–line variant a heuristic was described in [13]. An overview of the properties of the problem and a branch and bound method for solving the problem was proposed in [14]. Results of computational experiments on comparison of the branch and bound method and a heuristic were reported. The experiments used the integer programming model and the IBM ILOG CPLEX package.

This article deals with problem (1)–(4) with restrictions (a) and with forbidden zones for the case when \( G \) is a bipolar directed graph. A local search polynomial–time algorithm is proposed. In practice, the partial order between objects can describe the sequence of processing parts on various technological equipment.

3. Algorithm for search of local optimum
3.1. Statement of subproblems

Introduce the following definition.

Denote by \( D(s, t) \) an directed graph that consists of two or more chains going from the vertex \( s \) to the vertex \( t \) and having no other common vertices except \( s \) and \( t \). Moreover, on each of the chains connecting \( s \) and \( t \) all vertices (except \( s \) and \( t \)) have a second degree. We will call the graph \( D(s, t) \) a bipolar directed graph (BDG).
An example of the BDG is shown in Figure 1.

Denote the range of admissible solutions of problem (1)–(4) by $B$. Range $B$ is disconnected and it consists of $r$ separate blocks $B_k$ of length $L_k$ that must contain the placed objects, $B = \bigcup_{k=1}^{r} B_k$. Then an admissible solution to problem (1)–(4) corresponds to some partition of objects into blocks. An example of the object placement area for the case $r = 2$ is shown in Figure 2.

![Figure 1. Example of the BDG.](image)

**Figure 1.** Example of the BDG.

We will call an admissible solution to the problem (1)–(4) a *local minimum* if the goal function on it takes the smallest value when the partition of objects into blocks be fixed which satisfies the given partial order between objects.

Let the partition of objects into blocks be fixed. Denote by $I_k$ the set of object numbers in the block $B_k$; $n_k$ is the capacity of the set $I_k$. Since $I_k \cap I_l = \emptyset$ for every $k, l = 1, \ldots, r$, then to find the local optimum of the problem it is sufficient to find the minimum in $r$ independent subproblems. In every block $B_k$ it is possible to consider the subproblem of placement $n_k + 2$ objects. In $B_k$ the subproblem contains $n_k$ placed objects with numbers from $I_k$ and two imaginary objects $s$ and $t$ which correspond to the left and the right borders of $B_k$ with coordinates $LB_k$ and $RB_k$ respectively. Denote by $I_L(B_k)$ and $I_R(B_k)$ are the sets of objects to the left and to the right of the block $B_k$; $J_L(B_k)$ and $J_R(B_k)$ are the sets of zones to the left and to the right of $B_k$ respectively.

In the case when the placement area on the line is bounded on the left and on the right by zones, we consider the arbitrary block $B_k$. Let $G$ is the BDG. Denote by $G' = (V', E')$ the weighted graph of connections between objects in $B_k$. With the preceding notations we note, that $V' = I_k \cup \{s, t\}$.

Since the graph $G$ has the BDG structure, then the set $I_k$ can be represented as a union of non–intersection chains. Let $G'$ has $a$ chains, and $c_i$ denote the number of vertices of chain $i$ in block $B_k$, $i = 1, \ldots, a$. For simplicity of notation, let the set $I_k$ includes the following numbers...
of vertices of the chains \( I_k = \{(1, \cdots, c_1); (c_1 + 1, \cdots, c_1 + c_2); \cdots; (c_1 + c_2 + \cdots + c_{a-1} + 1, \cdots, c_1 + c_2 + \cdots + c_a)\} \). Define the set \( E' \) as follows.

(i) If the vertices \( i, j \in I_k \) and the arc \((i, j) \in E \) then the arc \((i, j) \in E' \).
(ii) Draw the arcs \((s, i) \in E', i = \{1, c_1 + 1, \cdots, \sum_{y=1}^{a-1} c_y + 1\} \), from the vertex \( s \) to the initial vertices of the chains of \( G' \).
(iii) Draw the arcs \((j, t) \in E', j = \{c_1, c_1 + c_2, \cdots, \sum_{y=1}^{a} c_y\} \), from the final vertices of the chains to vertex \( t \) of \( G' \).

Let \( u'_{ij} \) is the weight of the arc \((i, j) \in E' \). Denote by \( r(i) \) the next vertex to the right from the vertex \( i \) in \( G \), and denote by \( l(i) \) the previous vertex to the left from the vertex \( i \) in \( G \). Let \( S_y \) be the number of the last vertex on the chain with the number \( y \). Then \( S_y = \sum_{i=1}^{y} c_i \), and we assume that \( c_0 = 0 \). We define the weights of the arcs in the graph \( G' \) as follows.

(i) \( u'_{ij} = u_{ij} + \sum_{p=1}^{S_y} \sum_{q \in J_L(B_k)} w_{pq} + \sum_{p=1}^{l(i)} \sum_{q \in J_R(B_k)} w_{pq} \),

for \( \forall i, j \in I_k, (i, j) \in E, \quad b = 1, \cdots, a; \)

(ii) \( u'_{si} = u_{s(i)} + \sum_{p=S_{b-1}+1}^{S_b} \sum_{q \in J_L(B_k)} w_{pq} \),

where \( b = 1, \cdots, a, \quad i \in \{1, c_1 + 1, \cdots, S_{a-1} + 1\} ; \)

(iii) \( u'_{jt} = u_{j(r(j))} + \sum_{p=S_{b-1}+1}^{S_b} \sum_{q \in J_R(B_k)} w_{pq} \),

where \( b = 1, \cdots, a, \quad j \in \{c_1, c_1 + c_2, \cdots, S_a\} \).

Let \( l'_{ij} \) is the length of the arc \((i, j) \in E' \). Define the lengths of the arcs in the graph \( G' \) as follows.

(i) \( l'_{ij} = \frac{l_{ij} + l_{ij}'}{2} \), for \( \forall i, j \in I_k, (i, j) \in E; \)

(ii) \( l'_{si} = \frac{l_i}{2} \), for \( \forall i \in I_k, (s, i) \in E' ; \)

(iii) \( l'_{jt} = \frac{l_j}{2} \), for \( \forall j \in I_k, (j, t) \in E' . \)

Thus, defining the set of vertices and the set of weighted arcs and the set of arc lengths, we constructed the graph \( G' \) in the block \( B_k \). We assume that graphs \( G \) and \( G' \) are connected.

Below we present some properties of the graph \( G' \).

Note, that if \( G \) is the BDG, then constructed graph \( G' \) also is the BDG. By construction, \( G' \) differs from \( G \) in block \( B_k \) in arcs \((s, j), (i, t) \) which are added to the set of arcs \( E \) for some \( i, j \in I_k \). Other arcs are not added to the graph \( G' \) and the structure of the graph is preserved. So, by construction \( G' \) is also the BDG. In [11] for such graph of connections between objects the polynomial–time algorithm was proposed. Further we briefly describe this algorithm.

### 3.2. Optimal placement objects in the block

The algorithm for finding the optimal placement of the BDG vertices on the line consists of the following steps [11].

1. On each of the chains going from vertex \( s \) to vertex \( t \) we find an arc with minimum weight. If there are several such arcs, choose arbitrary one.
2. The graph \( G' \) is cut into rooted trees \( TL \) (left tree) and \( TR \) (right tree) in the arcs of minimum weight (step 1). The tree \( TL \) with the root in the vertex \( s \), the tree \( TR \) with the root in the vertex \( t \). The orientation of the arcs in the tree \( TR \) is reversed.
3. The optimal placement of the vertices of the trees \( TL \) and \( TR \) is independent [1]. This takes into account the weight of the arcs on which the cut is made.
4. We form the optimal placement of the vertices of the graph \( G' \) as follows: first we place the vertices of the tree \( TL \), and then the vertices of the tree \( TR \) in the reverse order.
3.3. Search of local optimum

For the case when $G$ is the BDG, the algorithm for search of local optimum is proposed. This algorithm consists of the following steps.

1. We find the partition of the objects into blocks, which satisfies the given partial order of placement of objects on the line.
2. Using the procedure described in 3.1, we consistently construct graphs $G'_1, \ldots, G'_r$ for fixed the partition of objects into blocks.
3. Using the algorithm from 3.2, for fixed the partition of objects into blocks, sequentially in each block we find the optimal placement of objects.

Computational complexity of the algorithm of optimal placement of the graph vertices in each block is $O(n \cdot \log n)$ [11]. The complexity of the local search algorithm is $O(m \cdot n \cdot \log n)$.

4. An example

Let the graph $G$ is not the BDG and let it has the form as shown in the Figure 3. For simplicity, let all vertices of the graph be numbered. The vertices of the graph correspond to the numbers of placed objects. Note, that the degree of vertex 3 is 3. The arc $(3, 7) \in E'$ will be called a jumper between the chains in $G$.

![Figure 3. Graph $G$ is not the BDG.](image)

Let the partition of the objects into blocks be fixed and $I_1 = \{1, 2, 3, 5, 6\}$, $I_2 = \{4, 7, 8\}$. Note, that vertices 3 and 7, which are forming the jumper in $G$, are in different blocks in such partition of the objects into blocks.

Consider the block $B_1$. There are two possible cases of passing the connection between vertex 1 with forbidden zone with number 1.

Case a). If $w_{11} > 0$ (see Fig. 4).

Note, that by construction $G'_1$ the value $w_{21}$ is already added to $u'_{s1}$ and $u'_{12}$, and $w_{51}$ is already added to $u'_{s1}$ and $u'_{15}$. As a result, the graph $G'_1$ is the BDG.

Case b). If $w_{11} = 0$, then during construction $G'_1$, $u'_{s1} = w_{21} + w_{51} > 0$. As a result, the graph $G'_1$ has the form as shown in Figure 4 and also it is the BDG.

For graph $G'_2$ the cases are considered similarly. Thus, $G'_1$ and $G'_2$ are the BDG and the local optimum of the problem can be found by the polynomial–time algorithm which proposed above.

So, if the graph $G$ is not the BDG and if the vertices $i$ and $j$, forming the jumper in the graph $G$, be in different blocks when constructing the partition of the objects into blocks. Then in each of these blocks the graph $G'$ will be the BDG. Therefore, the local optimum of the original problem can be found by polynomial–time algorithm.
5. Discreteness of the problem
Given a feasible placement, a remainder in the block $B_k$ is a segment of non-zero length between two adjacent elements (objects, zones) that do not have a common border or between the border of $B_k$ and an adjacent block [13].

In [13] it was proved that for a feasible solution $x$ to problem (1)–(4), we can find another feasible solution $x'$ for which there is not more than one remainder in every block and $F(x') \leq F(x)$. Thus, the original continuous problem is reduced to discrete problem for the case when $G$ is an undirected graph [13, 14].

Let $G$ is the BDG. We consider where will be the remainder in the block in the local optimum of the problem. So, the resulting placement of the vertices of $G'$ is combined as follows. First, the vertices of the tree $T_L$ are placed close to each other. Then the vertices of the tree $T_R$ are placed close to each other in the reverse order. As $T_L \cap T_R = \emptyset$ then the remainder in the block can be only between vertices $T_L$ and $T_R$. According to the polynomial–time algorithm of search of local optimum, in blocks $B_1$ and $B_r$ the remainder will be to the right of vertex $s$, close to $s$, and to the left of vertex $t$, close to $t$ respectively.

6. Conclusion
This article deals with the One–Dimensional Space Allocation Problem with the additional conditions. The structure of connections between objects is determined using the bipolar directed graph. A local search polynomial–time algorithm is proposed.

As the perspective for further research of this problem, we can specify search for different structures of the graph of connections between objects, for which it is possible to construct polynomial–time algorithms.

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 1973 Optimal linear ordering SIAM J. Appl. Math. 25 (3) 403–423
[2] Foulds L R, Hamacher H W, Wilson J M 1998 Integer programming approaches to facilities layout models with forbidden areas Annals of Operations Research 81 405–417
[3] Love R F, Wong J Y 1976 On solving a One–Dimensional Space Allocation Problem with Integer Programming INFOR 14 (2) 139–143
[4] Mukhacheva E A, Zalgaller V A 1993 Linear programming cutting problems Intern. J. of Software Engineering and Knowledge Engineering 3 (4) 463–476
[5] Panyukov A V 2001 The problem of locating rectangular plants with minimal cost for the connecting network Diskret. Anal. Issled. Oper. 8 (1), ser. 2 70–87
[6] Picard J C, Queyranne M 1981 On the One–Dimensional Space Allocation Problem Oper. Res. 29 (2) 371–391
[7] Rudnev A S 2009 Probabilistic tabu search algorithm for the packing circles and rectangles into the strip
$Diskret. Anal. Issled. Oper.$ $16 (4)$ 61–86

[8] Sarkar A, Batta R, Nagi R 2006 Placing a finite size facility with a center objective on a rectangular plane
with barriers $European Journal of Operational Research$ $179 (3)$ 1160–1176

[9] Simmons D M 1969 One-dimensional space allocation: an ordering algorithm $Oper. Res.$ $17 (5)$ 812–826

[10] Zabudsky G G 2005 On the complexity of the problem of placement on a line with restrictions on minimum
distances $Russian Math. (Iz. VUZ)$ $49 (12)$ 9–12

[11] Zabudsky G G 2000 On the problem of the linear ordering of vertices of parallel-sequential graphs
$Diskret. Anal. Issled. Oper.$ $7 (1)$ 61–64

[12] Zabudskii G G, Amzin I V 2013 Algorithms of compact location for technological equipment on parallel lines
(in Russian) $Sib. Zh. Ind. Mat.$ $16 (3)$ 86–94

[13] Zabudskii G G, Verenchuk N S 2016 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 DOI: 10.1134/S1990478916010154

[14] Zabudsky G G, Verenchuk N S 2018 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
https://doi.org/10.1007/978-3-319-93080-43