GREEDY AND BRANCHES-AND-BOUNDARIES METHODS FOR THE OPTIMAL CHOICE OF A SUBSET OF VERTICES IN A LARGE COMMUNICATION NETWORK

Boris Melnikov
Department of Computational Mathematics
and Cybernetics,
Shenzhen MSU – BIT University,
China
bormel@smbu.edu.cn, bormel@mail.ru

Yulia Terentyeva
Department of Communication Network,
Center for Information Technologies and Systems
for Executive Authorities,
Russia
terjul@mail.ru

Article history:
Received 15.01.2023, Accepted 25.05.2023

Abstract

The problems of the proposed paper are generated by the actual tasks of communication networks. The development of communication resources is accompanied by an increase in the dimension of existing communication networks, for which the usual tools for solving network problems are becoming increasingly ineffective. At the same time, the spectrum of communication network problems covers both classical graph theory problems and specialized problems linking them with mathematical models of various fields of mathematics, including optimization theory, dynamic programming, probability theory and the theory of heuristic algorithms.

This paper is devoted to the study of the problem of optimal allocation of a certain resource on the objects of the communication network. In this case, a complete set of specialized equipment at the nodes of the communication network is considered as a resource. It is necessary to optimize the number of pieces of equipment with the condition that all fiber-optic communication lines of the network are under the control of installed reflectometers. To solve this problem, variants of two methods were described: the greedy algorithm and the method of branches and boundaries. On the basis of the described algorithms, the computer programs were implemented and computational experiments were carried out; in the latter, the dimension of the communication network was chosen high enough to guarantee the legality of using the selected variants of algorithms for real communication networks. A representative series of experiments has shown that it is more expedient to use variants of the greedy heuristic algorithm for the group of problems under consideration, this paper contains a detailed argumentation of the result obtained.

The considered problem can also be transferred to cases of optimal placement of other resources on the objects of the communication network. At the same time, if there is a specific objective function, there will be another formalization of restrictions. But the approaches to choosing the optimum may be identical, which makes the results obtained in the paper more important due to their participation in potential generalizations of the problem.

Key words

Communication network, heuristic algorithms, optimal subset selection, graph theory models, greedy algorithm, ranches-and-boundaries method.

1 Introduction

At the beginning of the paper, we repeat some sentences from the recent paper [Melnikov and Terentyeva, 2022] about the existing *connection of the tasks we are considering with physical data layer transmits bits over physical communication channels*. This connection of those problems that can be called related to physics with the problems of the theory of communication networks is also fully traced in this paper.

Thus, the physical data layer transmits bits over physical communication channels, such as coaxial cable or twisted pair. That is, it is this level that directly transmits data. At this level, the characteristics of electrical signals that transmit discrete information are determined, for example, the type of encoding, data transfer rate, etc. And after that it is necessary to consider the control of the communication network, i.e., its various algorithms.

When designing a communication network, the calculation of its stability is mandatory requirement. In the case of large-scale communication networks, this becomes a problematic issue, which is caused by the NP complexity of algorithms for obtaining the correct stability estimate.

In this way, a completely different problem than in [Melnikov and Terentyeva, 2022] is considered, moreover, it is little related to it, if viewed from the point of view of the problems that arise for them in graph theory. However, these two problems are very closely related in terms of their practical application, and we have already briefly said about this.

The problems of the proposed paper are generated by practical work with communication networks. The modern pace of development of civilization and communication resources are inevitably accompanied by an increase in the dimension of existing communication networks, for which the usual tools for solving network problems are becoming increasingly ineffective. All these factors stimulate the development of new effective methods for solving problems on communication networks. At the same time, the spectrum of communication network tasks covers both graph theory close to classical problems and specialized problems synthesizing

- technical features of the design and operation of communication networks
- with mathematical models of various fields of mathematics, including:
  - optimization theory,
  - dynamic programming,
  - probability theory,
  - and the theory of heuristic algorithms.

This paper is devoted to the study of the problem of optimal allocation of a certain resource on the objects of the communication network. In this case, a complete set of specialized equipment (so called reflectometers) at the nodes of the communication network is considered as a resource. It is necessary to optimize the number of pieces of equipment with the condition that all fiber-optic communication lines of the network are under the control of installed reflectometers. The control condition is formalized. It should be noted that this optimal distribution problem was NP-hard, and, therefore, it requires the development of special heuristic algorithms, as well as the study of the effectiveness of these heuristic algorithms. The research was also dictated by the importance of the task, since for high-dimensional networks, which are real communication networks, the factor of optimizing resources, including equipment in the form of reflectometers, becomes one of the most significant.

To solve this problem, two groups of methods were programmed:

- the simple greedy algorithm, [Hromkovič, 2003];
- and the branches-and-boundaries method, [Hromkovič, 2003; Hromkovič, 2004] etc.

The dimension of the communication network was chosen high enough to guarantee the legitimacy of using the chosen most effective method for real communication networks.

A representative series of experiments has shown that it is advisable to use a greedy heuristic algorithm for the optimal placement of reflectometers on communication network objects for the purpose of total control of fiber-optic communication lines. This paper contains a detailed argumentation of the obtained result.

2 On the relation with the previous works of the authors on this topic

Thus, for the organization of a specific communication network:

- we consider a fairly relevant problem of reflectometry,
- propose a graph model for it,
- give some possible algorithms for solving the problem within the framework of the proposed model,
- give brief results of the first computational experiments conducted,
- and draw preliminary conclusions about the possible application of these algorithms.

More specifically, in the paper we consider the topic of reflectometry of optical fibers of the communication network, related to the identification of sections of fiber-optic cable that can be changed as a result of internal or external destabilizing factors. Within the framework of this topic, a specific task is considered, i.e., the problem of optimal placement of reflectometers on network objects.

The paper continues the subject of works: [Melnikov, 2005; Melnikov, 2009; Melnikov et al., 2021] etc. We shall use the terminology described in these works without further explanation, as well as the terms introduced directly into them:

- “anytime algorithm”,
- “subtask array”,
- “right subtask”,
- “left subtask”,
- “sequence of right subtasks”,
- etc.

More precisely, we continue to consider the application of the branches-and-boundaries method (its extension, previously titled *a multi-heuristic approach*) in a lot of discrete optimization problems. At the same time, we add several auxiliary heuristic algorithms to the “usual” variants of the branches-and-boundaries method (BBM), which are almost equally implemented in various subject areas. However, unlike all other subject areas, here in the process of computational experiments we have not received any acceptable gain from the use of BBM; as a result, we make a preliminary conclusion that the simplest (greedy) algorithms are more successful in the subject area under consideration.

Let us repeat only the description of two the above-mentioned concepts “anytime algorithm” and “sequence of right subtasks”. Anytime algorithms are real-time algorithms that have the best (at the moment) solution at each specific moment of operation, the user can view these pseudo-optimal solutions (also in real time), and the sequence of such solutions gives usually in the limit the optimal solution.

Apparently, more detailed definitions are not needed. But here is a possible *example* of the practical application for such anytime algorithms; this example, by the way, is suitable for any difficult problem. Thus, let us estimate the time to complete the entire task in 3 months, and the customer (the boss, the user) wants to get *at least some* acceptable solution much sooner; of course, it is desirable that it is more or less close to the optimal one. To say, he wants to obtain the first solution after 1 hour of the program. If the total time to complete the entire task is significantly more than 1 hour, then some solutions that are close to optimal begin to be obtained “almost” in 3 months, to say, in 2 months and 29 days. What should we do? That is why we should use some additional heuristics, which give even a “more distant” solution. (comparing the usual BBM), but very quickly. One of these heuristics (in fact, a modifications of BBM) is the use of so-called 1-trees. The other is so-called *right tasks sequence*, see Fig. 1.

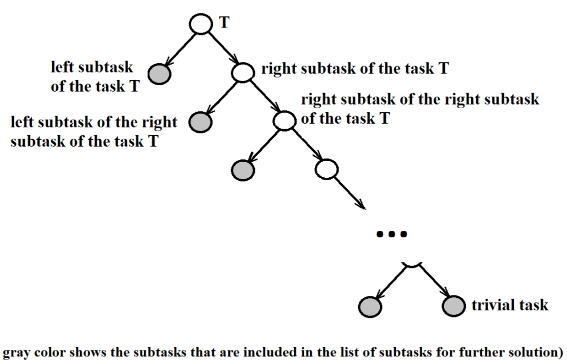

Figure 1. The organization scheme of the sequence of right subtasks.

This heuristic is as follows. Each time we select the next separating element in some subtask (let it be T; we can also say “when getting the next right problem”) we actually construct such a sequence when applying the heuristic:

- task T itself,
- the right (sub)task of the task T,
- the right task of the right task of the task T,
- etc.

Certainly, each time the corresponding left problems are constructed (and included in the list of problems for potential solution in the future):

- the left task of the task T,
- the left task of the right task of the task T,
- etc.

The described process ends:

- either when obtaining a trivial problem (for example, a zero-dimensional problem): in this case, we remember its solution (the boundary, the tour received at the time of its setting, and other *characteristics*) as *a pseudo-optimal solution of the current time* of the anytime algorithm;
- or when we get a sufficiently large boundary in any problem, for example, greater than the pseudo-optimal solution available at a given time.

Note that in practice, the described process of building a sequence of right subtasks *does not take much extra time* and *does not lead to a large increase of the list of tasks* intended for potential solutions in the future.

Thus, we have described a simple process of constructing an anytime algorithm based on some specific variant of BBM; it is in fact the *truncated* branches-and-boundaries method.

The value of finding the optimal solution for the problem under consideration also lies in the fact that this task can be extended to a number of emerging problematic situations related to the allocation of resources on communication network facilities. It is also obvious that when considering this problem with parameters characterizing the properties of reflectometers, as well as taking into account the topology of the communication network, we get a model of optimal equipment placement, which is close to the real one. The study of the problem of optimal equipment placement is ultimately important from the point of view of saving resources, which in the case of high-dimensional communication networks can be critical and thereby affect the most important technical indicators of the communication system.

3 Some similar formulations of the problems of reflectometry. Informal description

From the general descriptions of BBM algorithms, let us return directly to our problems. The relevance of the subject area under consideration is primarily due to the need to minimize the cost of so-called reflectometers; we also consider the existing restriction on the condition of total monitoring of fiber-optic cables. Solutions to specific discrete optimization problems arising in the subject area should contain:

- a subset of communication network objects on which reflectometers should be placed,
- as well as an indication of a specific type of reflectometer on an object whose technical characteristics are parameters affecting the formation of the aforementioned subset.

Similar problems arise when designing and / or upgrading a communication network, and they are especially important in situations where the communication network has a very large dimension. The search for possible practical algorithms for solving this problem, which is NP-hard, is the subject of this paper.

Thus, the purely engineering problem considered here has an essential mathematical component, since with large dimensions of the communication network, the search for an exact solution becomes impossible and requires research, first of all, approaches to the solution, including heuristic algorithms, synthesis of classical graph search algorithms, etc., as well as the development of the solution algorithms themselves with their approbation and computational experiments.

It is important to note that the mathematical formulation of the problem that arose from the problem of optimal placement of reflectometers on the communication network can also be used *for another, equally important network problem*, i.e., the problem of optimal placement of repair crews (and necessary spare equipment) at communication nodes; we propose to consider this slightly modified formulation in future publications.

Let us move on to the informal description of the discrete optimization problem considered in the paper. A communication network is given in the form of an undirected graph; at the same time, as is quite common in our models ([Bulynin et al., 2020; Melnikov et al., 2020] etc.), a pair of its coordinates is given for each vertex of the graph. In addition, some types of "lighting devices" are specified (hereinafter referred to as "lamps", and without quotation marks), and each type of lamp has power (measured in units of length) and cost; with a future solution, an unlimited number of lamps of each type can be used. We have to put some lamps in some vertices of the graph, in order to optimize the value of a special goal function (which we shall talk about later).

In all models considered in the paper, we believe that a lamp placed at a vertex illuminates all vertices located at a distance not exceeding the power of this lamp (among the illuminated vertices, of course, the vertex itself in which this lamp is placed). It is possible to place a lamp in a vertex (a lamp of the type we choose in the vertex of the graph we choose) in the process of solving the problem.

The complete statement of the problem includes a certain set of vertices of the graph, in which lamps are forbidden to be placed. According to the terminology used in our previous papers on the use of BBM in discrete optimization problems (some references were given above), such vertices can be called "taboo"; however, to be even more precise, this term should be attributed not even to some vertices, but to pairs (vertex, lamp), however, this seems to be unprincipled.

Besides, we note in advance that we did not use such restrictions during the computational experiments described below, although there is a possibility of such restrictions in our programs (data structures).

The task itself depends on the goal function; it usually consists in the fact that it is necessary to illuminate the maximum possible number of vertices, and among the lamp placements that illuminate such a maximum possible number of vertices, we need to choose a set of vertices that have a minimum cost. At the same time, numerous variants of the goal function are possible:

- for example, some linear combination of the number of vertices and the total cost of lighting: it is clear that in this case, such values are included in the objective function with different signs;
- or a variant that somehow depends on the execution time;
- etc.

Looking ahead, we note that any such option when describing the general algorithm of BBM and its software implementation can be a criterion for ordering existing subtasks in an array of pointers to these subtasks.

4 The formal graph model

The terminology of graph theory is consistent with [Harary, 1969; Diestel, 1997; Gera et al., 2016; Karpov, 2017; Gera et al., 2018]. We shall give a formal model for one of the possible simplified variants, when:

- there are no vertices of the graph in which lamps are forbidden to be placed;
- all vertices need to be illuminated;
- the target function is selected as a minimum of the cost of the lamps used.

Problem statement.
Input:

- undirected graph $G = (V, E)$; assume that
$$V = \{ v_1, v_2, \dots, v_m \}, \quad E = \{ e_1, e_2, \dots, e_n \}.$$
- 2 coordinates for each of the vertices of the set V, defining the vertex of the graph as a point of the unit square, i.e.
$$(\forall i \in \{ 1, 2, \dots, m \}) \, (X(i) \in [0, 1], \; Y(i) \in [0, 1]);$$
we shall represent the coordinates in the program as integers, from 0 to 10^6 (where 10^6 corresponds to the value 1);
- k types of lamps; for each lamp there is a power (lighting length) $\mathrm{Dlin}(i)$ and a cost $\mathrm{Cost}(i)$; both these functions are set for each $i \in \{ 1, 2, \dots, k \}$.

Auxiliary definition of the distance between vertices v_i and v_j: the distance is the minimum length of the path along the edges of the graph between these vertices, where the edge length is calculated in the usual geometric way:

$$\rho(v_i, v_j) = \sqrt{(X(i) - X(j))^2 + (Y(i) - Y(j))^2} \, ;$$

this formula is true when $(\exists e \in E)(e = \{v_i, v_j\})$, but even if such a condition is false, the distance between the vertices v_i and v_j (we defined it before) will be denoted in the same way, i.e. $\rho(v_i, v_j)$.

Intermediate solution: for $i = \{1, 2, \ldots, m\}$, such solution is the mapping

$$K : i \to \{0, 1, 2, \ldots, k\};$$

in this case, any positive number (as the value of K) means the number of the lamp placed at the vertex, and 0 means the absence of such lamp.

The cost of an interim solution:

$$\sum_{i \in \{1,2,\ldots,m\}} \mathrm{Cost}(K(i)).$$

Acceptable solution: such is any intermediate solution in which all vertices are illuminated; vertex v is illuminated if and only if

$$(\exists i \in \{1, 2, \ldots, m\}) \, (\rho(v, v_i) \leqslant \mathrm{Dlin}(K(i)));$$

we allow the option $V_I = V$.

Output: (pseudo-) optimal placement (as an acceptable solution), for which the cost is minimized.

End of the problem statement description.

It is clear that the above formulation can be considered:

- both as a problem of finding an optimal solution;
- and as a a task of finding a pseudo–optimal solution (for example, in the presence of predefined time constraints).

5 Greedy algorithm for solving the problem

The simplest goal function of the greedy algorithm, which is used in the algorithm below, is to estimate the number of newly illuminated vertices.

Formulation of the greedy algorithm.

Input, output: coincide with the input and output data of the formulation of the problem given in previous section.

Method.

Step 0. Initialization: as an intermediate solution, we set the mapping $K(i) = 0$ for each $i = 1, 2, \ldots, m$.

Step 1. If the considered intermediate solution is an acceptable solution, then exit the algorithm with it as the answer (output).

Step 2. Choosing some i from the set $i \in \{1, 2, \ldots, m\}$, such that $K(i) = 0$; for it (for this i), choosing some j from the set $j \in \{0, 1, 2, \ldots, k\}$; here, the choice of the pair (i, j) is carried out in such a way as to maximize the value $N/\mathrm{Cost}(j)$, where N is the number of *newly illuminated* vertices when placed in the i-th vertex of the j-th lamp.

Step 3. If there is no possibility of choosing a pair (i, j) at Step 2, exit the algorithm with the available intermediate solution as the answer.

Let us remark, that we leave this step in the description of the algorithm, although for such a description of the problem statement and the algorithm under consideration, such a lack of choice is impossible. And therefore, in our version, the answer will always be some *acceptable* solution (and not just an intermediate solution). However, all this is possible in other similar situations.

Step 4. Adding the selected pair to the intermediate solution (i.e., assigning $K(i) = j$) and going to Step 1.

End of the algorithm formulation.

In the remaining part of the paper, we consider the goal function to be given (i.e., we believe that it is formulated at Step 2 of the above algorithm) and we shall not specifically talk about its definition; however, we shall consider its implementation.

6 Variants of solving the problem by the branches-and-boundaries method

Let us firstly remark, that in the implementation of the branches-and-boundaries method and in the process of computational experiments, we considered some following variants.

- The variant that does not use BBM, i.e., in fact, just a greedy algorithm considered before.
- The variant "with a simple BBM", in which it is possible to assign $K(i) = j$ (for some possible i and j, according to the condition of the problem); this assignment for the problem under consideration:
  - is performed in its right subtask,
  - is prohibited in its left subtask,
  - and at the same time in the right subtask, it remains unchanged until the end of its processing.

  Remark that according to the terminology of our previous papers, this assignment in the left subtask is the so-called taboo resolving element.
- The variant "with a complicated BBM", in which we allow the possibility of subsequent replacement of the lamp with a brighter one (but not with a less bright one). such a replacement makes sense *on this branch of computing*.
- The variant "with a very complicated BBM", in which we allow the possibility of subsequent replacement of the lamp with both a brighter and a less bright one. Remark that such a replacement may indeed make sense: in the process of forming an intermediate solution, a situation may arise when everything "nearby" is already lighted, and at the same time, unlike the results of preliminary calculations performed using greedy heuristics, a "less bright" lamp at the vertex in question is sufficient..

Note that here, when implementing (or when presenting data structures), there are three variants (not two variants, as almost always) for having a lamp for the future step of BBM:

- "there exists" (such a lamp),
- "missing" (such a lamp),
- "prohibited" (such a lamp).

(Let us also repeat the brief description of all the variants mentioned in the introduction. We can say that we are considering some variants of BBM, the conditions of which can be ordered "from more hard", where an action once performed cannot be canceled, at least on the branch of considered calculations, to "less hard".)

Now, we give a formal description of one of these BBM-algorithms.

Formulation of the simple variant of the branches-and-boundaries method

Comment: in the formulation of the BBM-algorithm described here, we *do not use* the auxiliary algorithm for constructing a sequence of right subtasks, which we have already mentioned above.

Input, output: coincide with the input and output data of the formulation of the problem given in Section 4.

Method.

Step 0. Initialization consisting of the following actions.

- Create a list of subtasks consisting of a single subtask; in this initial subtask, we set the mapping $K(i) = 0$ for each value $i \in \{1, 2, \ldots, m\}$; note that this mapping can also be considered as an intermediate solution.
- Create the current pseudo–optimal solution coinciding with such an intermediate solution.

Step 1. If the list of subtasks is empty, then exit the algorithm with the answer that is the current pseudo-optimal solution. (In other versions of the method, when some other stop condition is met. First of all, such a possible stop condition is exceeding the pre-set time limits, but many other options are also possible.)

Step 2. Selecting the first subtask from the available list (and excluding it from the list). We shall call the selected subtask the current one, denoting it, if necessary, T.

Step 3. On the basis of the subtask T, building its right subtask (in a new memory segment), let it be T_r. A one-time modification of the subtask T as *its* left subtask, let it be T_l. The right subtask is built on the basis of the greedy heuristics described above, and, similar to the algorithm from section Section 5, is applied at this step once. (At the same time, each of the two new subtasks, i.e., T_r and T_l, may turn out to be "degenerate", and in this case, it will not be considered at the next steps of the algorithm.)

Step 4. If the task T_r has a "small dimension", then we implement its final solution made by the brute force method. If at the same time the obtained solution is better than the current pseudo-optimal one, then we replace the current pseudo–optimal solution with the newly obtained one. Otherwise, if the "small dimension" is not observed, then we add T_r to the list of subtasks; at the same time, the ordering of this list goes according to the criterion, which is a special modification of the greedy heuristic.

Let us remark, that in relation to the problem we are considering, "the small dimension" means a small number of vertices that have not yet been illuminated. At the same time, the specific definition of a "small number" depends on the specific implementation of the algorithm.

Step 4′. For the subtask T_l instead of T_r, repeat Step 4 and going to Step 1.

End of the algorithm description.

In conclusion of the section, we note that here, due to the limitations on the volume of the paper, we do not have the opportunity to describe in detail the additional heuristics *added to the above general description* of the branches-and-boundaries method. Therefore, we shall here list them very briefly; thus, these are the following possibilities.

- Preliminary formulation of *several different* variants for choosing a separating element, [Melnikov, 2009; Melnikov et al., 2021]. After that, choosing one of them, for directly using in the BBM-method, by additional application of risk functions, [Melnikov and Radionov, 1998], including dynamically generated ones.
- Clustering situations (in other words, clustering subtasks), [Wang et al., 2009; Agichtein et al., 2012; Blundell et al., 2012; Blundell and Yee, 2013; Mehrotra and Yilmaz, 2017] etc. That means, in some cases, the partial rejection of a detailed search for a separating element using a greedy algorithm and choosing instead of such a search for a separating element found earlier in a "similar" situation.
- Preliminary "sorting" of graph elements, [Melnikov et al., 2020; Melnikov et al., 2021].

Let us add the two important remarks; each of them applies to some above items and can be considered as some "preprocessor actions".

- About time of the execution of the algorithm. In our previous publications, we have repeatedly noted that such actions *cannot "worsen"* the execution of the algorithm: we do not lose the ability to apply any other separating elements on other branches of computing. However, such actions can sometimes "improve" the execution of the algorithm. Of course, it is impossible to "prove" theoretically anything like that, but in practice the time gain is often noticeable.

- About "similar situations". Here, the best option would be an auxiliary algorithm for such an arrangement of graph vertices (points), in which the total distance between neighboring vertices (plus the distance from the last to the first) would be minimal. However, this is a pure traveling salesman problem (TSP), it is NP-hard, so of course it cannot be used as an "auxiliary" one. And therefore we sometimes use some fast heuristic auxiliary algorithms for such an initial arrangement.

Some additional explanations of the algorithms discussed here can be found further in the description of the software implementation. Moreover, we propose to publish the approach to software implementation in a separate paper.

7 A possible approach to generating input data for computational experiments

This section discusses algorithms for generating input data for conducting computational experiments: unlike most of the previously considered applied discrete optimization problems, here such generation is one of the main components of the entire problem under consideration (and the important part of the subject of this paper).

There is such a broad meaning of the concept of "representativeness"; it characterizes those parameters that will provide the most adequate result of the analysis of the sample population. In a broad sense, the concept of representativeness borders on the measure of correspondence between the general population and the sample, how accurately this sample describes the features of the general population under study. At the same time, representativeness is a measurable value, it can be determined by the so–called representativeness error, i.e., the difference between the specially selected characteristics of the sample and the general population. The final quantitative value of representativeness is usually chosen as the quadratic mean of its possible values obtained.

However in practice, the actual value of this difference usually remains unknown. Therefore, a very difficult task is modeling structural objects and at the same time ensuring the representativeness of the model.

To study the behavior of structures and algorithms for their processing, modeling of the components of combinatorial structures based on random generation of a set of parameters is usually carried out. At the same time, the representativeness of the generated structures should be checked using statistical criteria specially adapted to these tasks. An adequate model of the structure generates objects that are close to real ones. The degree of approximation is also a separate subject of study of the modeling process of various processes and subject areas.

Let us return directly to the problems of reflectometry we are considering. Apparently, the title of this section could be replaced with a stricter one: from our point of view, we are considering *the only* possible approach to representative generation of test data for such tasks. At the same time, in most of the other tasks we considered earlier ([Melnikov, 2009; Melnikov et al., 2021] etc.), we paid significantly less attention to the representativeness of test data generation. Generalizing, we can say that representative data generation considered in this section *should not be used* in the two following situations:

- either the concept of representativeness (or the adequacy of the model) is *the subject* of the work, first of all, when this concept is put in the title; see the classical papers [Levin, 1984; Gurevich, 1984] and also [Abbott et al., 2011; Chasalow and Levy, 2021] etc.;
- or problems with deliberately "non–generatable" (including "one-time") input data are being investigated, which are taken exclusively from external sources, [Melnikov et al., 2021] etc.; as a result, the concept of continuous generation of input data in them can hardly make any sense.

We specifically repeat that in all other situations we consider representative generation of input data, similar to the one described in this section, necessary.

Thus, our task does not apply to the listed groups of problems. The fact that the approach is really the only possible one for our problems is explained by the complete impossibility of predicting in advance in these problems:

- not only how these data themselves,
- but even possible characteristics–invariants of these data.

On the other hand, if there is a possibility of such "predictions" (i.e., the possibility of successful use of such characteristics), then we can, in particular, obtain a set of tests close to real special cases of discrete optimization problems, and already on these sets to investigate the comparative characteristics of the algorithms being developed for solving such problems, [Cormen et al., 2009] etc.

As we said before, we propose to publish the approach to software implementation in a separate paper.

In conclusion of the section, let us talk about the *specific* algorithms used to generate input data that we currently use:

- all vertices are initially supposed to be available for any kind of lamps;
- as already noted, we set 4 types of lamps; each type has its own "brightness" and cost.

8 Some results of computational experiments and conclusion

This section summarizes the results of computational experiment, and it is worth noting that in reality, the temporary improvement of BBM compared to the greedy algorithm has never exceeded 6 %. This section is also the conclusion (and, of course, the conclusion of the paper, but not the topics about the problems of reflectometry): in it, we present preliminary outcome about the comparative effectiveness of the algorithms considered for the problems of reflectometry.

The computational experiments carried out by us can be briefly described as follows.

- We generate graphs of "medium saturation", and at the same time, as we have already written, new edges are added first of all "to more or less close vertices". That is, everything is done approximately as in the computational experiments of many of our previous topics related to the use of random graphs, [Bulynin et al., 2020; Melnikov et al., 2021] etc.
- The specific constants we used: dimensions 29 and 99; 4 types of lamps.
- We *manually* selected the cost and power of these lamps, in such a way that the greedy algorithm described before for the vast majority of generated special cases of the problem (95% of cases and more for such "medium saturation graphs") would use all 4 types of lamps.
- As a result, this greedy algorithm for dimension 99 processes all generated special cases of the problem in less than 1 minute each.

We used a "medium power" computer. Specifically: `Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz`.

Two of the randomly generated special cases of the obtained problem are as follows (for the before mentioned dimensions 29 and 99 and the parameters of the generation function `pNT->InitRnd(2,3,0.7);`). We manually selected the values of powers (lengths) equal to 4 types of lamps respectively:

$$40 \cdot 10^4,\ 18 \cdot 10^4,\ 8 \cdot 10^4 \text{ and } 2 \cdot 10^4.$$

(Let us repeat that we denote the lengths of the sides of the unit square as 10^6, and therefore, for example, the value $40 \cdot 10^4$ actually represents $40 \cdot 10^{4-6} = 0.4$.) As we already mentioned, When applying these values, the greedy algorithm almost always advise to the use of all 4 types of lamps. Then the "median" examples we have obtained for the above two dimensions are as follows (the sample of output text of the program is given):

```
dim=29,
number of illuminated=29,
cost=196
used lamps by type:
1-1 2-2 3-2 4-5
```

and

```
dim=99,
number of illuminated=99,
cost=311
used lamps by type:
1-2 2-3 3-6 4-2
```

It can also be noted in addition to the above, that a significant increase of the dimension of the problem does lead neither to a significant increase in the number of lamps in the answer, nor to a significant increase the cost of this answer.

(It is also worth noting that the use of the greedy algorithm almost coincides with one of the auxiliary algorithms used in the multiheuristic approach to discrete optimization problems, which is a possible extension of the branches-and-boundaries method, see Section 2 of this paper and also [Melnikov, 2009] etc. Namely, we mean the construction of the sequence of right subtasks, see before. We already noted that the use of this auxiliary algorithm does not slow down the time of BBM; this thing can be explained theoretically and shown in practice. However, it is clear that in practice it is always desirable to use simpler algorithms.)

Let us return directly to the problems of reflectometry. It is the use of all 4 types of lamps that gives grounds to talk about the repeatability of the input data, and, consequently, to draw *preliminary* conclusions about the use of similar algorithms for much larger dimensions. As we have already said, in our calculations for the branches-and-boundaries method, there was practically no improvement in performance compared to the greedy algorithm. In connection with what was said in this section about constructing a sequence of the sequence of right subtasks, BBM did not have any time delays (they did not exceed 10% compared to the greedy algorithm), but there was no significant improvement in the quality of the solution: for example, for dimension 99 and some dozen examples considered, the maximum cost improvement the decision has never exceeded 6%. Note also that above, we have actually given all the important fragments of the program, and therefore it is very easy to repeat our calculations (with the above lamp values); the results should be almost the same as our ones. Therefore, we do not provide more detailed calculation results.

Thus, the obtained temporary improvement in the average operating time of this algorithm in the applied problem we have considered, compared with the greedy algorithm, *is very small*. This allows us to draw preliminary conclusions that *in the problems of reflectometry, the use of the simplest greedy algorithms is sufficient*, and we should not use more complex algorithms, in particular, branches-and-boundaries method. ("A negative result is also a result", attributed to T. Edison.)

At the very end of the article we note, that the considered problem can also be extrapolated to cases of optimal placement of other resources on the objects of the communication network. At the same time, of course, there will be another formalization of constraints in the presence of an objective function. But *the approaches to choosing the optimum may be identical*, which thereby makes the results obtained in the paper more valuable due to their participation in future potential generalizations of the problem.

References

Abbott, J., Heller, K., Ghahramani, Z., and Griffiths, T. (2011). Testing a Bayesian measure of representativeness using a large image database. *NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 2321–2329.

Agichtein, E., White, R., Dumais, S., Bennet, P. (2012). Search, Interrupted: Understanding and Predicting Search Task Continuation. *Proceedings of the 35th International ACM SIGIR conference on Research and development in information retrieval*, pp. 315–324.

Blundell, Ch., Yee Whye Teh, and Heller, K. (2012). Bayesian Rose Trees. arXiv:1203.3468 [cs.LG].

Blundell, Ch. and Yee Whye Teh (2013). Bayesian hierarchical community discovery. *Proceedings of Advances in Neural Information Processing Systems*, pp. 1601–1609.

Bulynin, A., Meshchanin, V., Melnikov, B., and Terentyeva, J. (2020). Algorithms for designing communication networks using greedy heuristics of various types. *CEUR Workshop Proceedings, 2667*, pp. 273–276.

Chasalow, K. and Levy, K. (2021). Representativeness in Statistics, Politics, and Machine Learning. arXiv:2101.03827v3 [cs.CY]

Cormen, T., Leiserson, Ch., Rivest, R., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, Massachusets.

Diestel, R. (1997). *Graph theory*. Springer-Verlag, Heidelberg.

Gera, R., Hedetniemi, S., and Larson, C. (2016). *Graph theory. Favorite Conjectures and Open Problems – 1*. Springer, Berlin.

Gera, R., Hedetniemi, S., and Larson, C. (2018). *Graph theory. Favorite Conjectures and Open Problems – 2*. Springer, Berlin.

Gurevich, Y. (1984). A logic for constant-depth circuits/ *Information and Control*, **61**, pp. 65–74.

Harary, F. (1969). *Graph theory*. Addison-Wesley Publ., Massachusetts.

Hromkovič, J. (2003): *Theoretical Computer Science. An Introduction to Automata, Computability, Complexity, Algorithmics, Randomization, Communication, and Cryptography*. Springer, Berlin.

Hromkovič, J. (2004): *Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics*. Springer, Berlin.

Karpov, D. (2017). *Graph theory*. Publishing House of the St. Petersburg Branch Mathematical Institute named after V. A. Steklov of Russian Academy of Sciences, Saint Petersburg (in Russian).

Levin, L. (1984). Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, **61**, pp. 15–37.

Mehrotra, R. and Yilmaz, R. (2017). Extracting Hierarchies of Search Tasks & Subtasks via a Bayesian Nonparametric Approach. arXiv:1706.01574 [cs.IR]

Melnikov, B. and Radionov, A. (1998). On the choice of strategy in non-deterministic antagonistic games. *Programming (Russian Academy of Sciences)*, **5**, pp. 55–67 (in Russian).

Melnikov, B. (2005) Discrete optimization problems – Some new heuristic approaches. *Proceedings – Eighth International Conference on High-Performance Computing in Asia-Pacific Region, HPC Asia 2005, 1592253*, pp. 73–80.

Melnikov, B., Tsyganov, A., and Bulychov, O. (2009). A multi-heuristic algorithmic skeleton for hard combinatorial optimization problems. *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization, CSO 2009, 5193637*, pp. 33–36.

Melnikov, B., Meshchanin, V., and Terentyeva, Y. (2020). Implementation of optimality criteria in the design of communication networks. *Journal of Physics: Conference Series, 1515(4), 042093*.

Melnikov, B., Dudnikov, V., and Pivneva, S. (2021). Heuristic algorithm and results of computational experiments of solution of graph placement problem. *Communications in Computer and Information Science, 1204 CCIS*, pp. 157–166.

Melnikov, B. and Terentyeva, Y. (2022). An approach for obtaining estimation of stability of large communication network taking into account its dependent paths. *Cybernetics and physics, 11(3)*, pp. 145–150.

Wang, Zh., Tang, X., Liu, J., Ying, Zh. (2009). Subtask Analysis of Process Data Through a Predictive Model. arXiv:2009.00717 [cs.HC]