The problem of pseudo-optimal placement of a graph on a plane

B.F. Melnikov[1], **V.A. Dudnikov**[2]

[1] Russian State Social University, Wilhelm Pieck str., 4, Moscow, Russia, 129226
[2] Togliatti State University, Belorusskaya str., 14, Togliatti, Russia, 445020

Abstract. The paper is devoted to the problem of placing a graph. Graphs provide an opportunity to present information in a visual and easy to understand form, so the problem of developing various algorithms for automatic placement of graphs on the plane is very relevant. In this paper, we propose a generalized mathematical model of the problem that allows us to consider the placement problem in n-dimensional space as the problem of finding a permutation of n elements. Based on the mathematical description "Hebene", an heuristic algorithm is built. Computational experiments were carried out on all pairwise nonisomorphic connected graphs up to and including 9. The algorithm found the optimal solution in more than 50% of cases, and in other situations the algorithm also produced acceptable solutions.

Keywords: graph problem, location problem, hard problem, heuristic algorithm.

1. Introduction

It is well-known, that a large number of graph problems are hard to solve [1]. This is primarily due to the fact that to solve them it is often necessary to go through an exponentially large number of variants, which usually makes it impossible to complete a search. For example, if we consider the general form of the traveling salesman problem with 25 vertices based on its statement from [2], then at the sampling rate of 35 million variants per second, all the decisions will take the same time as they took place after the Big Bang (about 13.7 billion years). And the problems that arise in real applications, in fact, may not be 25, but many thousands vertices. Therefore, it is obvious that an exhaustive approach to solving problems will rarely succeed.

Earlier, the problem of placing the graph was considered by other authors, for example, in [3, 4, 5]. In these works, algorithms of evolutionary modeling and other stochastic algorithms (genetic algorithms, other variants of evolutionary algorithms, the algorithm of simulated annealing [6], etc.) were used, and they have some success. However, some badness is seen in these approaches:

- the inability to control the process of the algorithm (seing its stochastic nature);
- the high complexity of a formal description of the class of problems, for which the set of parameters being considered (for example, for a genetic algorithm) obtains solutions that are close to the optimum;
- the strong dependence of the search capabilities on the temporal complexity of the algorithm: we can set the parameters for the genetic algorithm, that the optimum will still be found, but the operating time will be close to the time of the full search algorithm (the "brute force method").

In connection with the shortcomings listed above, the authors propose an approach that is heuristic, but not stochastic (which eliminates the use of such badness).

2. The formal statement of the problem

In this section, we consider the formulation of the problem and propose some new terms for describing the general problem of the placement of a graph. The term "graph placement model" introduced by us is used by the authors throughout the work. The graph placement model contains the following elements:

- the graph;
- the set of positions in which you want to place the vertices of the graph;
- a function that formally transfers a natural number (i.e., a position number) to a position from the set of positions;
- a goal function ([2] etc.).

Also using the model, some classes of the placement problems are described (for example, the considered class of problems of placing a graph in the plane). The description of the placement problem proposed below generalizes the statements proposed earlier in [3, 5, 7].

Definition 1. The model of placing the graph is called a quad

$$\mathcal{M} \;=\; < G, M_\eta, H, f > \;,$$

where:

- $G =< V, E >$ is a graph (V is its set of states, E is its set of edges);
- M_η is the set of allocations of a multidimensional space; remark that we use in the paper the Greek letter η to denote the dimension of the graph (i.t., the dimension the space where the graph is placed); and the Latin letter n (which is "traditional" for such situations) is used to denote the number of vertices of the graph;
- considering X as the set of all the placements (i.e., X is the set of x such that $x =< x_1, x_2, \ldots, x_n >$, where $x_i \neq x_j$ for $i \neq j$), the function H is $H : X \to M_\eta$;
- $f : M_\eta \to \mathbb{R}^+$.

We also introduce some following terms which are based on [1, 2].

- the signature of the model is a triple $< M_\eta, H, f >$;
- the variant of the model is determined by some subset of all possible inputs; i.e., the model with the signature given on it is a variant of the model;
- an instance of the problem is specified on a variant of the model with a specific graph.

Examples of the model variant are:

- the problem of placing the vertices of a graph on the plane "into the grid";
- the problem of graph visualization, which is described in much the same way: the points of the coordinate plane are numbered, and the number of intersections of edges in the graph is chosen as the goal function;
- the problem of placing elements of integrated circuits; in the simplest case, this problem, according to [8], coincides with the problem of graph visualization.

And the following example of a model of the graph visualization problem will be considered in more detail. Signature for this model can be the triple $< M_2, I, g >$, where:

- M_2 is the set of allocations of the 2-dimensional space;

- the function I is $I : X \to M_2$;
- $g(x)$ is the function counting the number of intersections of edges in the considered placement.

Figure 1 depicts an instance of the graph visualization problem. The dashed line marks the grid into which the placement takes place, the position numbers and their coordinates are marked in the corners of the grid.

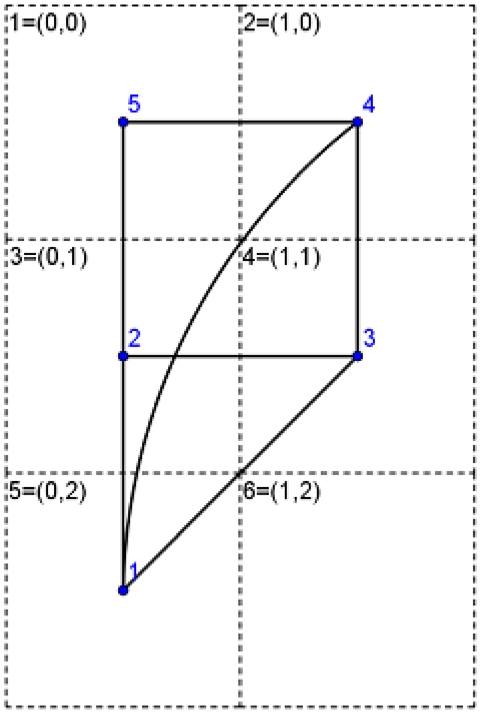


Figure 1. Example of an instance of the graph visualization problem.

Thus, if $x = \{5, 4, 2, 3, 1\}$, then $I(x) = \{(0; 0), (1; 0), (0; 1), (1; 1), (0; 2)\}$, and $g(I(x)) = 1$.

The model $< G, M_1, H, f >$ will be called the model for placing the graph in the scale, and

$$\mathcal{L} =< G, M_2, H, f >$$

will be called the model for placing the graph in the grid (on the plane).

The problem of minimizing the placement of a graph. Such a task consists in minimizing the function of the model f. If we consider the previous version of the model for the graph visualization problem, the task is to minimize the number of intersections of edges, i.e., in finding a placement

$$x^* = \{1, 2, \ldots, n\}, \text{ such that } g(I(x^*)) \to \min_{x \in X} g(I(x)).$$

More formally, the problem can be described as follows.

Thus, we have given an instance of the placement problem. We need to find the placement

$$x^* =< x_1^*, x_2^*, \ldots, x_n^* >, \text{ such that } \big(\forall x \in X\big) \ \big(f(H(x^*)) \le f(H(x))\big),$$

where X is the set of permutations of the set $\{1, 2, \ldots, n\}$.

Next, we shall discuss model variants with the function $f(x)$ of the following type only:

$$f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a(x_i, x_j) \cdot d(x_i, x_j), \text{ where:}$$

- $a(x_i, x_j) = \begin{cases} 1, \text{ if } (x_i, x_j) \in E, \\ 0, \text{ if } (x_i, x_j) \notin E. \end{cases}$
- $a(x_i, x_j)$ is a metric; for instance, for 1-D real coordinate space, we can set $d(x_i, x_j) = |x_i - x_j|$.

The specific algorithms for implementing such a minimization problem can be realized, for example, using a multi-heuristic approach to the development of algorithms for solving discrete optimization problems described in our previous papers [6, 9, 10] etc.

3. An heuristic algorithm for pseudo-optimal placement of a graph

We describe with the help of block diagrams the algorithm of pseudo-optimal placement of the graph, called the acronym "Hebene" (from "Heuristics of best neighbor"). Figures 2 shows a flowchart of auxiliary `HebeneIter` procedure, it is the one iteration of `Hebene` algorithm. And a flowchart of the whole `Hebene` algorithm using the `HebeneIter` procedure is shown in both Figures 2 and 3.

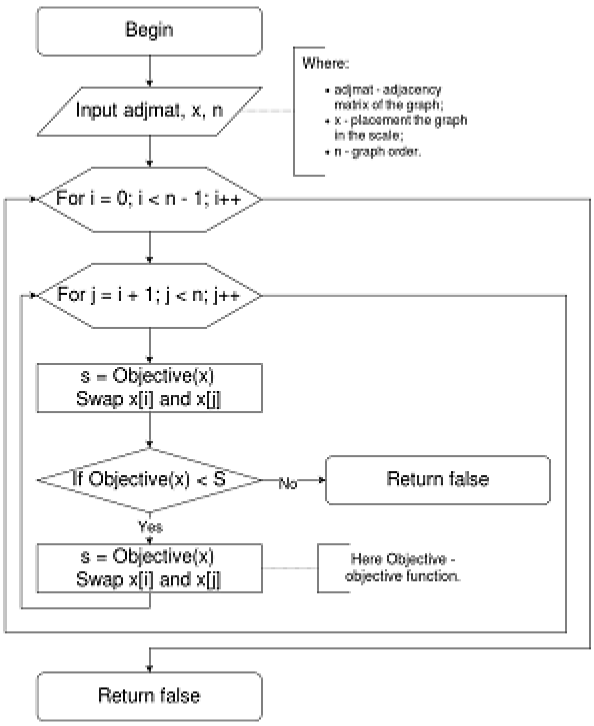


Figure 2. One iteration of `Hebene` algorithm.

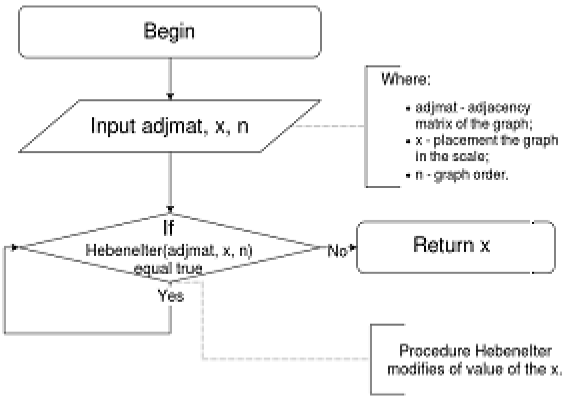


Figure 3. `Hebene` algorithm.

Thus, the algorithm depicted in the flowcharts is divided into two procedures. The first procedure exchanges all pairs of vertices, and if the value of the goal function is improved, then it returns `true`, otherwise `false`. The second procedure executes the first procedure until it returns `false`.

4. Some results of computational experiments

First of all, let us give an illustrative example of the work of `Hebene` algorithm for placing a graph on a plane. Figures 4 and 5 (see below) depict the same graph, but the graph on Figure 4 was chosen with random placement, and the placement for the graph in Figure 5 was found by the `Hebene` algorithm. The work of the algorithm was aimed at minimizing the number of intersections between the edges.

Now, we shall give the results of the computational experiments in more detail. For carrying out computational experiments, the authors considered all pairwise non-isomorphic connected graphs of orders from 3 to 9. (See the total number of connected pairwise non-isomorphic graphs for the calculated orders in [11]; for instance, there exist 6 such graphs for dimension 4, 853 such graphs for dimension 7 and 261080 such graphs for dimension 9.)

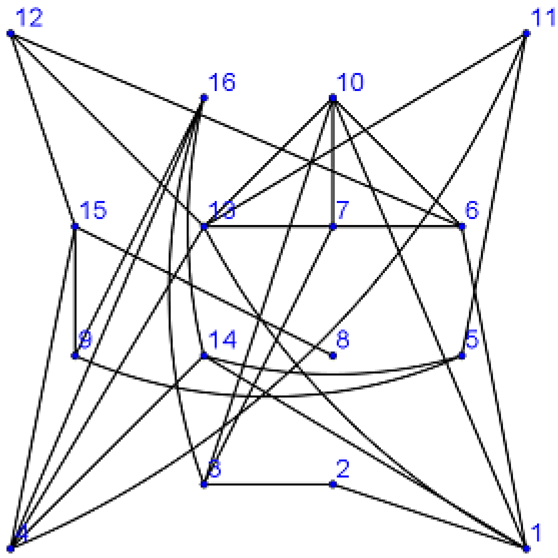


Figure 4. A random placement of a graph on the plane.

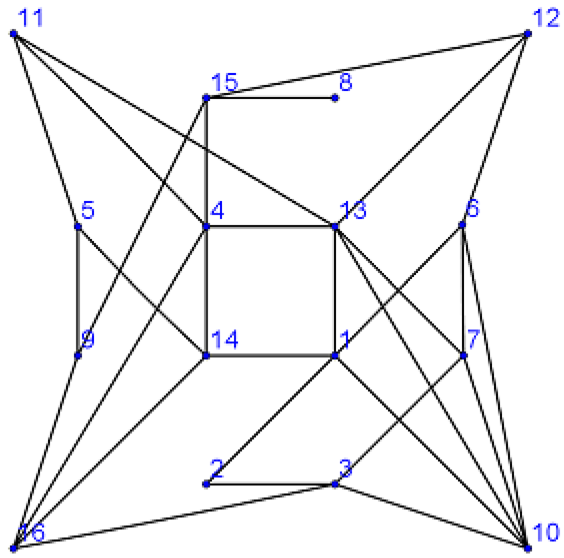
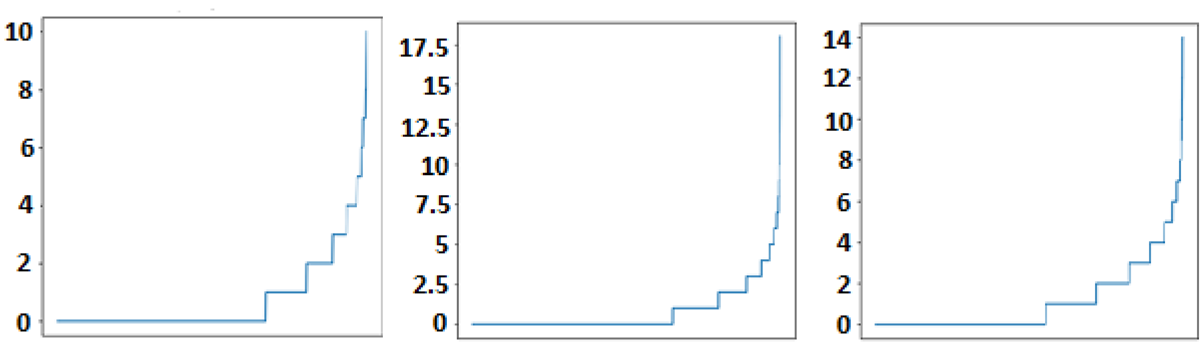


Figure 5. Optimized placement of the graph on the plane.

Further, for each graph, the minimum of the problem of minimizing the placement of a graph with considered before model $\mathcal{L}$ were computed *using a complete search algorithm*. In such a way, data were generated to the following comparing with the capabilities of the proposed heuristic algorithm.



$\Delta = |f(x^*) - f(x^h)|, \quad dim = 7$ $\Delta = |f(x^*) - f(x^h)|, \quad dim = 8$ $\Delta = |f(x^*) - f(x^h)|, \quad dim = 9$

Figure 6. Diagrams of comparisons for graphs of orders (dimensions) 7, 8 and 9.

From the diagrams of the comparisons there is clear, that they are similar to each other, namely, that the relative number of optimal solutions found by the Hebene algorithm is greater than 50% for the considered orders (from the 3rd to the 9th ones). Also, as can be seen from the graphs below, the difference between the optimal and the found by `Hebene` algorithm decreases with increasing number of graphs.

Such an analysis of the results of the conducted experiments shows the distribution of solutions found by the `Hebene` algorithm with respect to optimal solutions. However, it is worthwhile to pay attention to the quantitative evaluation of all graphs together. We take as an estimate the ratio of the total number of values of the valuation function for all optimal solutions to the number of values of the solution estimation function found by the `Hebene` algorithm. Namely,

$$O^h(G) = \frac{\sum_{g \in G} f(x^*)}{\sum_{g \in G} f(x^h)}, \quad \text{where:}$$

- G is the set of all pairwise non-isomorphic graphs;
- x^* is, as before, the optimal solution for a given graph;
- x^h is the solution, found using the `Hebene` algorithm.

The results of such a criterion for the graphs under consideration are given in Table 1.

Table 1. The results of applying the first quality criterion.

p	7	8	9
$O^h(G)$	0.964	0.973	0.961

This criterion shows how far the solutions are optimal for all graphs. That is, if the criterion value is 1, then the algorithm always finds optimal solutions.

Now, for greater clarity, we shall show the value of the criterion for the same graphs, but with respect to the solution chosen randomly, i.e.,

$$O^r(G) = \frac{\sum_{g \in G} f(x^*)}{\sum_{g \in G} f(x^r)},$$

where x^r is a randomly chosen solution for the graph g, [10, 11]. Table 2 gives the results of estimating $O^r(G)$ for the same set of graphs as above.

Table 2. The results of applying the second quality criterion.

p	7	8	9
$O^r(G)$	0.550	0.534	0.446

Among some other auxiliary problems, we solved the problem of estimating the *minimum possible* values of the goal function $\mu_{\min}(\eta, p)$. Such an estimating is needed, first of all, for the analysis of placement algorithms for their search capabilities and will be considered in more detail in subsequent work.

And now, we shall show the deviation of the estimate from the optimal values. Let us consider function

$$O^\mu(G) = \frac{\sum_{g \in G} \mu_{\min}(|V|, |E|)}{\sum_{g \in G} f(x^*)},$$

where $|V|$ and $|E|$ are quantities of vertexes and edges. In Table 3 below, we show the values of function $O^\mu(G)$.

The main problem with this estimate is that it takes into account only the number of vertices and edges, and does not account for the structure of the graph itself. Therefore, in the future, research will be aimed at finding an estimate that will take into account the structure of the graph.

Table 3. The results of applying the second quality criterion.

p	7	8	9
$O^\mu(G)$	0.850	0.822	0.732

5. Conclusions

Thus, computational experiments for the proposed algorithm were carried out *for all* pairwise non-isomorphic graphs up to order 9 inclusive (as mentioned earlier, due to computational complexity, not all graphs of order 9 were considered). During the computational experiments it was found that:

- the distribution of the values of the goal function for solutions found by the `Hebene` algorithm, is close enough to the distribution of the values of the goal function for solutions found by a full search (the "brute force method"); while the number of exact solutions found by `Hebene` exceeds 50%;
- the total value of the goal function for the solutions found by `Hebene`, differs from the optimal by no more than 4%, while this index for random solutions averages 50%;
- the total value of the estimate $\mu_{\min}(\eta, p)$ differs on average from the values of the goal function for optimal solutions by no more than 20%.

Based on this, it can be concluded that the algorithm has shown the good results and requires further research.

6. References

[1] Garey, M. *Computers and Intractability* / M. Garey, D. S. Johnson — USA: W. H. Freeman and Company, 1979. — 338 p.

[2] Hromkovič, J. *Algorithmics for Hard Problems. Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics* / J. Hromkovič — Berlin: Springer, 2004. — 548 p.

[3] Emelyanov, V. *Theory and practice of evolutionary modeling* / V. Emelyanov, V. V. Kureychik, V. M. Kureychik — Moscow: Fizmatlit. 2003. — 432 p. (in Russian).

[4] Balyuk, L. *Genetic algorithms for solving the problem of placing VLSI elements* / L. Balyuk // Izvestiya of Southern Federal University. Technical Science. — 2006. — Vol. 8 (63). — P. 65–71. (in Russian).

[5] Kostina, M. *Algorithms of artificial intelligence in the problem of graph visualization* / M. Kostina, E. Melnikova // Vektor Nauki of Togliatti State University. — 2014. — Vol. 3. — P. 31–35. (in Russian).

[6] Melnikov, B. *An approach to combining the unfinished branch-and-bound method and the algorithm of simulated annealing* / B. Melnikov, S. Eirich // Vestnik of Voronezh State University. — 2010. — Vol. 1. — P. 35–38. (in Russian).

[7] Dudnikov, V. *Genetic algorithm for solving the optimization problem of placing the vertices of a graph in the scale* / V. Dudnikov // Heuristic algorithms and distributed computations. — 2015. — Vol. 2. — P. 60–68. (in Russian).

[8] Lisyak, M. *Hybrid algorithm of multi-criterial placing VLSI elements* / M. Lisyak // Izvestiya of Southern Federal University. Technical Science. — 2012. — Vol. 7 (132). — P. 77–84. (in Russian).

[9] Melnikov, B. *Clustering situations in real-time algorithms for discrete optimization problems* / B. Melnikov, E. Melnikova // Control Systems and Information Technologies. — 2007. — Vol. 28. No. 2. — P. 16–20. (in Russian).

[10] Melnikov, B. *Applying multiheuristic approach for randomly generating graphs with the given degree sequence* / B. Melnikov, E. Sayfullina // Proceedings of higher educational institutions. Volga Region. Physics and mathematics. — 2013. — Vol. 3 (27). — P. 70–83. (in Russian).

[11] Harary, F. *Graphical Enumeration* / F. Harary, E. Palmer — N.Y.: Academic Press, 1973. — 216 p.