Application of artificial intelligence in the branch and bound method on the example of various applied problems

M.E. Abramyan[1], B.F. Melnikov[2], A.V. Nichiporchuk[3], M.A. Trenina[4]

[1]Southern Federal University, Bolshaya Sadovaya str. 105/42, Rostov-on-Don, Russia, 344006
[2]Shenzhen MSU – BIT University, 1 International University Park Road, Dayun New Town, Longgang District, Shenzhen, Guangdong Province, China, 518172
[3]Russian State Social University, Wilhelm Pieck str. 4, Moscow, Russia, 129226
[4]Togliatti State University, Belorusskaya str. 14, Togliatti, Russia, 445020

Abstract. The article describes the possible approaches to the use of artificial intelligence to improve the work of the branch and bound method in various applied problems. Various developments obtained by the authors earlier in the study of the branch and bound method are used, as well as new solutions to the problems are considered.

1. Introduction

The branch and bound method has been known since the middle of the XX century. Its active use is due to the need to replace the full search method. The solution of NP-complete problems by the method of full search can be possible only for certain special cases or in the case of a low dimension of the problem. Increasing the dimension causes a sharp increase in the number of steps of the algorithm. Currently, there are a large number of different methods that in one way or another allow you to get away from a complete search of all options.

The branch and bound method uses the idea of clipping subsets that knowingly do not contain optimal solutions. The process of finding a solution consists of two parts, branching and searching for estimates. Simplistically, the first part builds a tree of disjoint subsets of values of the target variable, and the search for estimates for each subset contains the upper and lower bounds of possible values of the target variable for a particular subset. Subsets with the worst scores (minimum or maximum, depending on the problem statement) are excluded from consideration. A detailed description of the branch and bound method can be found in [6].

This article will discuss the problems for which the authors have previously used the branch and bound method, as well as describe the possibilities of using artificial intelligence to improve the search for solutions.

2. Problem of reconstructing the distance matrix between DNA chains

The problem of reconstructing the distance matrix between DNA sequences arises in bioinformatics. The length of the DNA sequence makes it impossible to find the exact distance between the sequences, but the practical aspect of the problem allows finding a solution that is close to optimal (but perhaps not optimal).

The branch and bound method is used to reconstruct a distance matrix that allows for blank elements. In this case, to determine the quality of recovery, a characteristic of the matrix, called *badness*, is considered. In fact, badness shows how the triangles formed by all possible elements of the matrix differ from the elongated isosceles.

The badness of a triangle is calculated using the formula:

$$\sigma = \frac{a - b}{a},$$

where a, b, c are the sides of the triangle, with a ≥ b ≥ c. It is allowed to use a similar formula for the angles of the triangle. To find the badness of the matrix, the maximum badness value of all triangles, or their sum, is used.

The space of admissible solutions to this problem is all possible sequences for determining unknown elements of the upper triangle of the matrix. The separating elements that are used for branching at the next step of the method are empty elements. An auxiliary heuristic is used to select the separating element. A detailed description of the application of the method can be found in [1, 2, 3].

According to the research carried out in [1], the branch and boundary method gives an improvement in the badness index by about 20% compared to the recovery by known methods (for example, using the Needleman-Wunsch algorithm [4]).

3. Minimization of nondeterministic finite automata

The problem of vertex minimization of nondeterministic finite automata (NFA) is to find an NFA equivalent to the original one and at the same time containing the minimum number of vertices. The problem is NP-difficult, but at the same time has an important practical value [5] and often requires a solution in real time, when the received request algorithm gives the best solution at this step.

To apply branch and bound method in this problem, it is necessary to bring the problem to the matrix form. Two canonical automata with sets of states X and Y are constructed for the initial NFA. For subsets of these sets, the relation # is defined, which can be described as a bipartite undirected graph G, whose edges connect elements of sets X and Y. The graph G has no isolated vertices, and for any two vertices of the graph, the sets of adjacent vertices are different.

On the basis of the graph G, its adjacency matrix A is constructed, which sets the relation #. The elements of the set X correspond to the rows of the matrix, and the sets Y correspond to the columns. At the intersection of the elements for which the relation x # y is satisfied, the matrix contains 1, and all other elements are equal to 0.

The *grid* of a matrix A is a collection of rows and columns (possibly non-contiguous), with only ones at the intersection. A grid is called complete if it cannot be expanded by adding a new row or column. If the set of complete grids includes all the single elements of the original matrix, then we will call this set the matrix coverage, and the number of complete grids is the size of the coverage. The problem is thus reduced to finding the coverage of the original matrix, which has a minimum size.

Branching of the branch and bound method occurs by adding another grid to the set. The grid in this case is the dividing element. Two parameters are used to find the lower bound of subtasks, called *BoundSimple* (the number of ones in the current set of grids) and *BoundSecond*, defined by the formula:

$$\boldsymbol{BoundSecond = A * BoundSimple + B * Yes.Count + C * No.Count},$$

where *Count* is the number of elements in the sets *Yes* (full grids selected for this subtask) and *No* (grids that are excluded from consideration in this problem and its descendants). *A, B, C* are integer coefficients.

The results of numerical experiments have shown that the branch and bound method for minimization of NFA gives good results, and the modifications described in [7] allow these results to be improved even more.

4. Traveling salesman problem

The classical traveling salesman problem, repeatedly described in the literature, is formulated as follows in the form of a graph: there is a complete weighted graph G whose vertices correspond to

cities and whose edges correspond to roads between them. The weight of the rib is equivalent to the cost of transportation on the road. It is usually assumed that the graph is completely connected, otherwise missing roads correspond to edges with a weight significantly greater than the weight of other edges. To solve the problem, it is necessary to find a Hamiltonian cycle with a minimum weight.

The branch and bound method in this problem uses the edge of the graph as the dividing element and the sum of the edge weights as the estimate. Note that the adjacency matrix of the original graph is used for convenience of analysis.

Previously, the authors conducted statistical studies [8] for three different variants of the traveling salesman problem: geometric (the weight of the graph edge is equal to the distance between the vertices connected by this edge on the plane), pseudogeometric (the element of the distance matrix is multiplied by a coefficient selected from the vector of normally distributed numbers) and random (all elements of the matrix are generated randomly with a uniform distribution). The results obtained showed that for a pseudogeometric problem, the application of the branch and boundary method is effective, especially in combination with clustering situations.

5. Application of artificial intelligence

When solving the problems under consideration, a good result is obtained by a multi-heuristic approach [9], based on the idea of combining different heuristics, which separately demonstrated the improvement of the solution. However, the search for heuristics that can improve the results obtained is often "random" in the sense that it is based on observations and assumptions. Due to the large volume of processed data, it can be difficult for a person to immediately pick up a heuristic that will work.

The choice of heuristics is often based on a pattern in the source data, so it is possible to effectively use machine learning algorithms to find patterns. Depending on the formulation of the original problem, you must choose a suitable problem that can be solved by machine learning algorithms.

To improve the work of the branch and boundary method in the reconstruction of the distance matrix of DNA sequences, it is possible to additionally introduce a risk function into the solution of the problem. It is a function that corrects the weight of elements in badness depending on the pass number. The pass number must be taken into account, because each pass increases the number of elements recovered approximately. The risk function formula is as follows:

$$\boldsymbol{E} = \frac{\sum \boldsymbol{c_i E_i}}{\sum \boldsymbol{c_i}},$$

where c_i is some coefficients, and E_i is the value of the matrix element obtained on the i-th step.

The choice of coefficients can be made based on the solution of the regression problem. To do this, you need to calculate the problem with the same source data, changing the set of coefficients c_i. the Result of solving the problem is the value of the badness of the restored matrix. Thus, it is possible to determine the correspondence between the set of coefficients c_i and the badness of the matrix. Training the model on the set of such correspondences makes it possible to roughly determine the badness of a set of coefficients, without calculating the problem by the method of branches and boundaries. This will allow you to cut off obviously bad sets of coefficients and not waste time on the work of the branch and bound method.

Similarly, it is possible to do in the case of the task of minimizing the NFA. In this problem, there is a calculation of the BoundSecond parameter, in the calculation of which integer coefficients are involved. Here, the coefficients A, B, and C are taken as input parameters in the regression problem, and the number of complete grids required for coverage can be used as the result of the model.

In the case of a salesman's task, it is necessary to take into account the specific statement of the problem. If there is a "randomness" in the statement, the input parameters of the regression can be the distribution parameters.

In these situations, a problem arises when a regression model trained on one matrix is applied to another matrix. Most likely, such a step will give bad results. But it is too expensive to train the model for each matrix, since the training involves repeated application of the branch and bound method. In

this case, it is easier to calculate the original problem without using machine learning methods. This situation can be circumvented by initially trying to divide the original matrices into clusters.

Clustering is the division of a set of objects into groups in such a way that objects within one set are similar to each other, and objects in different sets are as different as possible. Clustering is well suited for the primary analysis of objects, because it does not require pre-calculated results and looks for patterns in a set of objects. Based on clustering, you can use a single regression model for matrices within a single cluster.

6. Conclusion

The considered approach using machine learning algorithms provides some basis for further research and search for heuristics that can improve the results of the branch and boundary method in the described problems.

7. Acknowledgments

The reported study was partially funded by RFBR, project number 19-31-90161

8. References

[1] Melnikov B. On possible methods for solving the problem of reconstructing the matrix of distances between DNA strings / B. Melnikov, M. Trenina // Fuzzy Technologies in the Industry - FTI Proceedings of the II International Scientific and Practical Conference, 2018. – P. 11-20.

[2] Melnikov, B.F. The application of the branch and bound method in the problem of reconstructing the matrix of distances between DNA strings / B.F. Melnikov, M.A. Trenina // International Journal of Open Information Technologies. – 2018. – Vol 6(8). – P. 1-13 (in Russian).

[3] Abramyan, M.E. Impleementation of the Branch and Boun Method for the Problem of Recovering a Distances Matrix Between DNA Strings / M.E. Abramyan, B.F. Melnikod, M.A. Trenina // Modern Information Technologies and IT-Esucation. – 2019. – Vol. 15(1). – P. 81-91 (in Russian).

[4] Needleman, S. A general method applicable to the search for similarities in the amino acid sequence of two proteins / S. Needleman, Ch. Wunsch // Journal of Molecular Biology. – 1970. – Vol. 48(3). – P. 443-453.

[5] Geldenhuys, J. Reducing nondeterministic finite automata with SAT solvers / J. Geldenhuys, B. van der Merwe, L. van Zijl // Finite-State Methods and Natural Language Processing. Lecture Notes in Computer Science. – 2010. – Vol. 6062. – P. 81-92.

[6] Goodman, S. Introduction to the Design and Analysis of Algorithms / S. Goodman, S. Hedetniemi – NY: McGraw-Hill, 1977. – 344 p.

[7] Abramyan, M.E. Minimization of nondeterministic finite automata: heuristics in the implementation of branch and bound method / M.E. Abramyan, B.F. Melnikov // Numerical algebra with applications. Proceedings of Eighth China-Russia Conference – Southern Federal University, I.I. Vorovich Institute of Mathematics, Mechanics, and Computer Science, 2019 – P. 102-106.

[8] Melnikov, B.F. Clustering of situations in solving applied optimization problems (on the examples of traveling salesman problem and distance matrix recovery) / B.F. Melnikov, A.V. Nichiporchuk, M.A. Trenina, M.E. Abramyan // International Journal of Open Information Technologies. – 2019. – Vol. 7(5). – P. 1-8 (in Russian).

[9] Melnikov, B.F. Multiheuristic approach to discrete optimization problems // Cybernetics and Systems Analysis. – 2006. – Vol. 42(3).