SOME SPECIAL HEURISTICS FOR DISCRETE OPTIMIZATION PROBLEMS

Boris Melnikov[1], Alexey Radionov
Department of Mathematics and Information Science,Togliatti State Univ., Belorusskaya str., 14, Togliatti, 445667, Russia

Viktor Gumayunov
Department of Telecommunication Software,Ulyanovsk State Univ., L.Tolstoy str., 42, Ulyanovsk,432700, Russia

Keywords: Anytime algorithm, discrete optimization problem, local heuristics, minimization, nondeterministic finite automata, disjunctive normal forms.

Abstract: In previous paper we considered some heuristic methods of decision-making for various discrete optimization problems; all these heuristics should be considered as the combination of them and form a common multi-heuristic approach to the various problems. And in this paper, we begin to consider local heuristics, which are different for different problems. At first, we consider two problems of minimization: for nondeterministic finite automata and for disjunctive normal forms. Our approach can be considered as an alternative to the methods of linear programming, multi-agent optimization, and neuronets.

1 INTRODUCTION

In previous papers (see (Melnikov, 2005), and also some papers in Russian) we considered some heuristic methods of decision-making for various discrete optimization problems (DOP). In fact, all these heuristics should be considered as the combination of them and form a common multi-heuristic approach to the various DOP. And in this paper, we consider not the common heuristics (which can be applied to various DOP), but so called local heuristics, which are own for different problems. Let us remark, that in (Melnikov, 2005) we already considered also some local heuristics (for a problem of considered in this paper, i.e., for minimization of automata).

But at first, let us briefly describe the common set of heuristics of (Melnikov, 2005). The object of each of considered problems is programming anytime algorithms.

- We use some modifications of truncated branch-and-bound method (B&B).
- For the selecting immediate step, we apply dynamic risk functions.
- Simultaneously, for the selection of coefficients of the averaging-out, we use genetic algorithms.
- And the reductive self-learning by the same genetic methods is used for the start of truncated B&B.

Thus, this combination of heuristics represents a special approach to construction of anytime-algorithms for the discrete optimization problems, which is an alternative to the methods of linear programming, multi-agent optimization, and neuronets.

2 CONSIDERED PROBLEMS

Thus, as we said before, the main object of this paper is local heuristics, and each of them belongs to its own area. Therefore let us describe considered DOP.

At first, we consider some connected problems of minimization for nondeterministic finite Rabin-Scott automata (NFA). Probably, the main for them is state-minimization, i.e., the problem of constructing NFA, which defines the given regular language and has minimum possible number of

[1] Author was partially supported by the Russian Foundation of the Basic Research (project No.04-01-00863).

Melnikov B., Radionov A. and Gumayunov V. (2006).
SOME SPECIAL HEURISTICS FOR DISCRETE OPTIMIZATION PROBLEMS.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - AIDSS*, pages 360-364
DOI: 10.5220/0002469803600364

states. Since (Kameda and Weiner, 1970), there are a few changes in description of the exact algorithms for this problem: all the algorithms are exponential relative to the number of states of considered NFA. The last argument is true because all the algorithms need to construct equivalent automaton of canonical form (or, maybe, some similar graphs or other objects). Let us remark, that from the point of view of the theory of complexity of algorithms, all the published algorithms (Kameda and Weiner, 1970; Jiang and Ravikumar, 1993; Melnikov, 2000; etc.) are equivalent. However we hope, that the approach of authors of this paper (Melnikov, 2000) allows formulating some heuristics for anytime algorithms.

Second, it is the problem of minimization of disjunctive normal forms (DNF). The exact algorithms for this minimization are obtained for ages (and are considered in the classical textbooks, for example, in the Russian textbook for first-year students (Yablonskiy, 1979), which is used more than 25 years), however the computer programs making on basis of such solutions cannot work in real time even for the number of variables, which is equal to 20, except, certainly, for a lot of trivial cases. The author does not know books, where any anytime algorithms for this problem are obtained, however, if such papers do exist, the approach of this paper, certainly, can be used in some alternative versions of computer programs.

In this paper, we shall consider local heuristics for two the described problems. However, let us briefly describe the two other problems, for which we are going to publish local heuristics in the next papers.

Thus, the third problem is the classical travelling salesman problem (TSP; see (Hromkovič, 2003), etc.); certainly, universal methods for solving TSP simply cannot exist. Some last years, authors of papers for heuristic methods of TSP-solution consider most often so called metric TSP. For their solving, some methods of linear programming and multi-agent optimization are used; (Hromkovič, 2003; Dorigo and Gambardella, 1997; Johnson and McGeoch, 1997; etc). However, some variants of the classical B&B can also be used not only for the exact (optimal) solution of considered TSP, but also for quasi-optimal heuristic solutions. At first, such approach can be used for the quasi-metric TSP (Melnikov and Romanov, 2001).

And the fourth problem is the special problem for graph transformation algorithms. Considering weighted oriented graphs, we formulate some special rules for combining their vertices. And the goal is to obtain graph having minimum possible number of edges. For details, see (Belozyorova and Melnikov, 2005) and the references from that paper.

3 SOME LOCAL HEURISTICS FOR THE NFA-MINIMIZATION PROBLEM

In this Section, we consider a heuristic algorithm for forming quasi-optimum covering; let it be Q defined in (Melnikov, 2000). For this thing, we shall select a subset of blocks (grids) of a matrix, which cells are corresponding to elements of special binary relation #, and this relation can be construct on the base of the given NFA. (See details also in (Melnikov, 2000).)

The considered algorithm is based on a special modification of truncated B&B. It differs from the classical truncated B&B that we do not divide the considered problem (and, therefore, the searching space) into the left and right ones. The whole searching space corresponds to the whole set of blocks, but for the first step, it is only the considered matrix of binary relation #.

But the practical programming for the classical truncated B&B gave the poor results. The main obstacle is that we hardly can fixed the fact that the considered block does not belong to the anytime solution. Besides, we can estimate only the cells, not the blocks (see such heuristics in (Melnikov, 2005)). And procedure of constructing blocks is the particular heuristic sub-problem; we are going to describe corresponding algorithms in the next paper.

Therefore, we have to use the following **modification of the truncated B&B**.

1. Considering the next problem of the searching space, we select a cell using the heuristics A (see below).
2. Using the heuristics B (see below), we construct the set of blocks M. Each of these blocks contains the selected cell. Let the number of blocks given by algorithm B be N.
3. The considered problem is divided into N ones. In each of obtained problem, we suppose that the next considered block belongs to Q, and other N-1 blocks do not belong to Q.
4. Returning to the step 1. (Or exiting, if the searching space is empty.) ■

This heuristic algorithm is based on the following example. Let us have the considered problem T, and after the heuristics A and B (see below), we obtain a set of blocks (let they be b1, b2 and b3) for the next branching. Then we divide the problem T for 3 ones (T1, T2 and T3, branching by b1, b2 and b3 correspondingly). And we use the fact that b1 could hardly be included in the set of blocks solving the problems T2 and T3, etc.

Heuristics A (selecting the cell). We select the cell having the maximum possible sum of the numbers of cells in the same row or in the same column, which are not included in the current answer. (Remark that in (Melnikov, 2005), we considered some more complicated heuristics for this thing.) ■

Heuristics B (selecting the block). Let us choose the set of blocks containing the cell of the row r and the column c.

1. First, let us construct the following quasi-block.
    a) Excluding columns, which have 0 in the position r. In the same way, we also exclude rows.
    b) Using algorithm A (or its modification of (Melnikov, 2005)) for estimating remaining columns and rows, such that the estimation is the sum of their values given by algorithm A. Certainly, we consider only cells, which values are equal to 1.
    c) Proving that there exist two columns or two rows, which have the different estimations. (Otherwise, we already obtained the quasi-block, i.e., all the corresponding cells have values 1. In this case, we add this quasi-block to the set of them and return to step b.)
    d) Excluding the row or the column having the minimum estimation.
    e) Returning to the step b.
2. For each constructed quasi-block, let us extend the number of rows and/or columns, such that all the corresponding values are equal to 1. Thus, we can obtain 1 or 2 blocks.
3. All the blocks constructed of the step 2 form the final set of blocks. ■

Certainly, for the successful execution, B&B needs not only heuristics for branching (i.e., heuristics A and B described before), but also some more heuristics:

- for calculating the bounds (C1 and C2);
- for the quick addition for the set of blocks (D);
- and for quick transformating the quasi-block into the block (E).

Let us remark in advance, that we do not describe heuristics E in this paper, and heuristics C2 will be described briefly. Their detailed description, and also the detailed description of some complicated heuristics which are the alternative to the simplest heuristics C1 given below, is the subject of the next paper.

Heuristics C1 (calculating upper bound). The upper bound is the maximum possible value of blocks, which are obtained for the ending of calculating the considered problem.

We count the number of rows which contain at least one value 1; also we count the number of such columns. The answer is the minimum of two these values. ■

Heuristics C2 (calculating lower bound). Similarly to C1, the lower bound is the minimum possible value of blocks, which are obtained for the ending of calculating the considered problem.

For calculating this value, we use another heuristics for constructing special set of cells, for which each their pair cannot belong to the same block. (Let us remark, that there exist, in general, more than 1 such sets of cells, but the mentioned heuristics constructs the only one.s) The number of such cells is the lower bound. ■

Certainly, the solved subproblem can be called unpromising if its lower bound is equal or more than minimum of the upper bounds of all the existing subproblems. Such subproblem can be excluded from the set of subproblems to be considered.

Heuristics D (the quick addition for the set of blocks). Two of the possible goals of applying this heuristics are the following: to make the upper bound; to make a sequence of the left problems (for the last thing, see (Melnikov, 2005)).

Thus, the simplest heuristics is the same as heuristics C1. ■

4 SOME LOCAL HEURISTICS FOR THE DNF-MINIMIZATION PROBLEM

Instead of blocks, we consider here the planes; each plane has dimension in the interval from 1 to the given number of variables N. At first sight, we have to construct all the planes, for which all the values of minimized function are equal to 1. (After this constructing, B&B can start.) However, all the well-known algorithms for constructing such planes are too long (Birkhoff and Bartee, 1999; Lee and Markus, 1967; Yablonskiy, 1979; etc) – unlike the NFA-minimization problem.

Really, if we use algorithms which obtain planes in decreasing order of their dimension, then we obtain that the time is $O(4^N)$. Let we have N variables and M sets of their values (corresponding to the sets of coordinates), where the minimized function is equal to 1. Then infilling the array corresponding planes (the number of planes is 3^N)

requires $O(M \cdot 2^N)$ units of time. Because at the worst $M = 2^{N-1}$, we obtain that the required time is $O(4^N)$.

Certainly, we exclude the planes belonging to other ones. The operation of such excluding require the time, which only linearly depends on N. But such procedure does not solve formulating problem completely.

Besides, if we use algorithms which obtain planes in decreasing order of their dimension, then the appearance of the first plane usually needs a lot of time; but it is a practical result, it hardly could be rigorously proven.

The algorithms which obtain planes in ascending order of their dimension also do not solve the problem, although the time estimating is here $O(3^N)$; this estimating is simply obtained, e.g., by realization algorithms of (Birkhoff and Bartee, 1999; Yablonskiy, 1979). This time estimating is some better, but also too long. However, the immediate start of B&B is here uninteresting, because there is unlikely that even the pseudo-optimal answer (pseudo-optimal DNF) contains planes having little dimensions.

However, the heuristics for the immediate start of B&B does exist. The detailed description of this heutristics is the subject of the special paper, we shall describe it briefly.

Thus, we set for this thing the following local goal: to construct the sets of considered subsets of the given planes, which intersections are minimum possible. The bound is here the power of intersection sets. And the indication of anytime decision is the absence of the sets of coordinates, for which the value of the given function is equal to 1.

Then we select an arbitrary value 1 from the given set of multidimensional cube corner. For it, we construct the plane containing it and having maximum possible dimension. (Such algorithm of constructing plane is similar to considered in Section 3 for the selecting block.) Certainly, we can obtain more than one planes. Then for each of these planes, we exclude then sets of coordinates belonging to this plane. It is important to remark, that each of them will form the different sub-problem, and, therefore, we can start truncated B&B before we have constructed the whole set of planes. Besides, such algorithm allow to obtain planes of big dimension, e.g., we often do not consider intersections of them.

We use here the following estimation for the bounds. The high bound is the number of planes in the quasi-optimum DNF (i.e., of the best DNF of the considered sub-problem). And the low bound is the same value for the considered DNF.

And, as we said before, the heuristics for minimization of DNF (unlike minimization of NFA) are given here very briefly. We are going to describe this thing more detailed in the next paper3.

5 SOME PRELIMINARY PRACTICAL RESULTS

While testing, we set the time for our anytime algorithm (see the tables). We also set the dimension of the problem – i.e., the number of rows for NFA (the number of columns depends of the last value also by special variate) and the number of variables for DNF – not the numbers of grids for NFA and planes for DNF, the last values are also special variates depending on previous ones.

The clock speed of the computer was about 2.0 GHz. If we choose the time under 10 minutes, we make the averaging-out by 50 or more solutions.

And the values of cells have the following meaning. For each cell, we made corresponding tests. For each test, we set the number of grids/planes for the given problem (certainly, we did not use this information in the program) and obtain the value of grids/planes found by anytime algorithm. Then we counted comparative improvement of this value (+) or the worsening (–). The possibility of positive values is the corollary of the fact, that, e.g. for the DNF, two planes of dimension k could form one plane of dimension k+1. The values were averaged; they are written in the table in percents. (I.e., +0.20 means that the mean value is better than the a priori given than 0.2%.)

Thus, below are the practical results.

NFA	20–23	40–45	60–65	80–90
01 sec	–1.76	–0.58	–0.02	–0.02
10 sec	–0.55	–0.17	+0.22	+0.20
01 min	–0.03	+0.45	+1.00	+1.06
10 min	0	+1.06	+1.07	+1.20
01 h	0	+1.07	+1.17	+1.21

DNF	20–22	25–27	30–33
01 sec	–8.5	–2.2	–1.9
10 sec	–1.21	–0.70	–0.43
01 min	–0.73	–0.65	–0.43
10 min	–0.03	–0.01	–0.01
01 h	–0.03	–0.01	0

Thus, the obtained results are near to 100%; this fact shows that the approach proposed in this paper could be applied in the future. And in the next papers, we are going to give the practical results for two other problems mentioned in Section 2.

REFERENCES

Belozyorova, A., and Melnikov, B., 2005. "Applying the set of heuristics in the problem of constructing scheme of nuclear transformations", 2nd Conference

"Methods and instrument of the information processing", Russia, Moscow State Univ. Ed. (2005) 208–212 (in Russian).

Birkhoff, G., and Bartee, T., 1999. Modern Applied Algebra, McGraw-Hill, N.Y., 1999.

Dorigo, M., and Gambardella, L., 1997. "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation, Vo.1, No.1 (1997) 53–66.

Hromkovič, J., 2003. Algorithms for Hard Problems, Springer, 2003.

Jiang, T., and Ravikumar, B., 1993. "Minimal NFA Problems are Hard", SIAM J. Comput., Vo.22 (1993).

Johnson, D., and McGeoch, L., 1997. "The Traveling Salesman Problem: A Case Study in Local Optimization", in "Local Search in Combinatorial Optimization", eds E.Aarts, J.Lenstra, John Wiley Ed., 1997, 215–310.

Kameda, T., and Weiner, P., 1970. "On the State Minimization of Nondeterministic Finite Automata", IEEE Trans.on Computers, C-19 (1970) 617–627.

Lee, E., and Markus, L., 1967. Foundation of Optimal Control Theory. Wiley, 1967.

Melnikov, B., 2000. "Once more about the state-minimization of the nondeterministic finite automata", The Korean Journal of Computational and Applied Mathematics, Vo.7, No.3 (2000) 655–662.

Melnikov B., and Romanov, N., 2001. "Once more on the heuristics for the traveling salesman problem", Russia, Saratov State Univ. Ed., "Theoretical informatics and its applying", Vo.4 (2001) 81–92 (in Russian).

Melnikov, B., 2005. "Discrete Optimization Problems – Some New Heuristic Approaches", Conference HPC-Asia-2005, IEEE Computer Society Press Ed., 2005.

Yablonskiy, S., 1979. Introduction into discrete mathematics, Moscow, Nauka Ed., 1979 (in Russian).