Analysis of Algorithms for Constructing Dense Sequencing of Digraphs Vertices

Valentina Turchyna[1][0000-0003-1051-9597], Kostiantyn Karavaiev[2][0000-0001-6062-4577],

[1]Oles Honchar Dnipro National University, D.Yavornitsky avenue, 35, Dnipro, 49000, Ukraine
vaturchina1949@gmail.com
[2]Oles Honchar Dnipro National University, D.Yavornitsky avenue, 35, Dnipro, 49000, Ukraine
karavaiev_k@fpm.dnulive.dp.ua

Abstract. In this work behavior of one exact polynomial algorithms for particular case of optimal sequencing problems when it's used for arbitrary graphs and sequencing widths is analysed. Based on this analysis several modifications of this algorithm are developed. Results of computational experiments has shown that proposed modifications are effective and substantially increase precision of finding optimal solution up to 98,5 percent. Additionally it is proved that an arbitrary optimal sequencing problem can be reduced to the problem with graph that has a dense sequencing for even sequencing width.

Keywords: discrete optimization, scheduling theory, optimal sequencing, approximate methods, level principle, maximum matching, dense sequencing

1 Introduction

When solving practical problems related to the optimal allocation of a finite set of jobs (tasks, projects, operations, etc.) between executors, two main classes are distinguished: the first - problems in which the order of execution is arbitrary, the second - problems in which technological constraints are imposed. This work is devoted to the latter. Since the technological constraints can be mathematically modeled with a directed acyclic graph, the problems under consideration are formulated as optimization problems on graphs. In general, these problems are NP-hard, so it is important to have effective approximate algorithms of polynomial complexity [1, 2].

2 Formulation of the problem

Consider one of the well-known problems of sequencing vertices of a directed graph. Suppose that a finite set of jobs is specified, on the order of execution of which the technological constraints are imposed [3-5]. Assuming that all jobs have the same execution time, the minimum time for which, without violating technological restrictions, all jobs can be executed by a given number of performers has to be determined.

It is natural to set technological constraints in a form of directed acyclic graph $G(V,U)$, $|V|=n$, where V is the set of vertices whose elements are associated with the jobs. Then the arrows correspond to the technological constraints. One of the optimization problems that arises in this case is the construction of optimal parallel sequencing, which is defined as such placement of the vertices of the digraph in line-arranged places for which the following conditions are satisfied:

- in each place there is no more than a given number of vertices, denoted h;
- if a pair of vertices $(i,j)\in U$, then vertex i is located to the left of vertex j;
- the number of non-empty places (called the sequencing length and denoted l) on which all vertices of the graph are placed is minimal.

In the general case, that is, for an arbitrary graph G and width h, this problem is NP-hard, so directional search schemes such as the branch and bound method are used to find the exact solution. Only for four partial cases the exact algorithms of polynomial complexity are found. This is the case when the graph G is directed forest [6] (this algorithm is called the level principle), when $h=2$ (two algorithms are known, one based on lexicographic order [7], the other one is being under the scrutiny in this work), when the graph G is opposing forest [8] and when the graph G is graph of bounded height [9].

3 Algorithm based on maximal matching

Suppose that we have an arbitrary acyclic directed graph G and $h=2$. In this case, in order to construct the desired optimal sequencing, one can use an algorithm that was historically first. This method of constructing S is called the algorithm based on the maximum matching.

Algorithm based on maximum matching.

1. For a graph $G(V,U)$, we construct an undirected graph $\overline{G}(V,E)$, where $(i,j)\in E$, if there is no path in the graph G neither from vertex i to j nor from j to i. Such a graph will be referred to as the reachability graph.
2. In the obtained graph $\overline{G}$, we find the maximal matching, which is denoted $M\subseteq E$, that is, a subset of the edges of the maximal cardinality with no vertices in common.
3. In the desired sequencing S^*, we consider all places empty and set $k=1$.
4. If G is empty, then the algorithm is finished.
5. One of the following cases is possible:
   (a) Among the open vertices of G, there exists one that does not belong to any of the edges in M, then we choose it to be placed.
   (b) In the set M there is an edge (i,j) such that vertices i and j are open, then we choose them for the placement and remove (i,j) from M.

(c) In the set M there is a pair of edges (i,p) and (j,q) such that vertices i and j are open, and between vertices p and q there is an edge in the graph $\overline{G}$, then we choose vertices i and j for placement, edges (i,p) and (j,q) are removed from M, and the edge (p,q) is added to M.

6. We place the selected vertices on the k^{th} place of the sequencing S^* and remove them from the graph G together with the arrows directed from them, if any. Set $G := G$, $k := k+1$ and go to 4.

It is important to note that polynomial complexity algorithms, such as the blossom algorithm, are known for finding M for arbitrary undirected graphs.

It was proved in [10] that the sequencing obtained by this algorithm is optimal. It is not straightforward to apply this algorithm to arbitrary sequencing width h due to the specificity of operations in step 5 of the algorithm.

4 A class of graphs for which dense sequencing exists

Consider some of the statements and their corollaries, which will narrow down a set of parallel sequencing tasks that need consideration, namely, to the class of graphs for which dense sequencings for an even h exist. This will allow us to generalize the algorithm based on the maximum matching for the case of arbitrary input data.

To the class of graphs for which there are dense sequencings, D_h, belong such graphs that, for a given value of the width h, have sequencings in which at each place there are h vertices.

Theorem 1. If there is an exact algorithm A of polynomial complexity for some fixed sequencing width $\tilde{h}$, then there is a polynomial algorithm for any $h' < \tilde{h}$.

Proof. Suppose that there is some non-empty graph G with n vertices and some fixed value $h' < \tilde{h}$. It is clear that the length of its optimal sequencing l^* is in the range from 1 (if $n \le h'$ and all vertices are isolated) to n (the graph is a chain) inclusive.

We construct a new graph G', which is obtained from the graph G by adding $(\tilde{h} - h')$ chains of some length $l \in \overline{1,n}$ to it. We now construct the sequencing S of the graph G' for $h = \tilde{h}$ by Algorithm A. It is known that it will be optimal and can be obtained in polynomial time.

It is clear that if $l = l^*$, then the length of S will also be l^*. On the one hand, it cannot be greater than l^*, since we have $(\tilde{h} - h')$ chains occupying $(\tilde{h} - h')$ positions on each of the l^* places, and vertices, belonging to the graph G can be placed on l^* places in case of $h = h'$. On the other hand, it cannot be smaller than l^*, since the chains can only be placed in a line and their length is l^*. Thus, if we remove from S all vertices belonging to the chains, then we obtain the optimal sequencing of S^* for

the original problem in polynomial time, since the addition and removal of chains can be done in linear time.

Therefore, if we can find the length of chains $l = l^*$, which is generally unknown, in polynomial time, then we can use algorithm A to find the optimal sequencing for $h' < \tilde{h}$. This can be done through binary search.

Note that $l = l^*$ corresponds to the minimum value of l for which there will be $\left(\tilde{h} - h'\right)$ vertices of the chains on all places in the resulting sequencing S obtained by the algorithm A for the graph G' and $h = \tilde{h}$. Indeed, for $l > l^*$, the length S will coincide with l, since it is necessary to place the chains, and the vertices of the graph G can be placed on a smaller number of places. For $l < l^*$, S will have places where there is less than $\left(\tilde{h} - h'\right)$ vertices belonging to the chains, otherwise vertices of graph G could be placed at a smaller number of places than l^* for $h = h'$.

We apply binary search to this problem. We choose some integer value of $l_0 \in \overline{a_0, b_0} = \overline{1, n}$, which splits this interval approximately in half. We construct a graph G', for that value of l. By applying algorithm A we find the optimal sequencing S for $h = \tilde{h}$. If there are $\left(\tilde{h} - h'\right)$ vertices belonging to the chains on all the places of S, then we choose $l_1 \in \overline{a_1, b_1} = \overline{1, l_0}$, and continue our search using this interval. If S has places where there are less than $\left(\tilde{h} - h'\right)$ vertices of the chains, then we choose $l_1 \in \overline{a_1, b_1} = \overline{l_0 + 1, n}$ and similarly continue the search. The algorithm convergence follows from the convergence of binary search for monotonic functions. The search requires logarithmic time, the check can be performed in linear time, and therefore we find the $l = l^*$ in polynomial time.

Since all the steps of the described algorithm can be performed in polynomial time, their number is finite, algorithm A has polynomial complexity under the condition of theorem, and thus described algorithm is exact and has polynomial complexity for any $h' < \tilde{h}$. ■

Theorem 2. If there is an exact algorithm A of polynomial complexity for some fixed sequencing width $\tilde{h}$ and graphs from $D_{\tilde{h}}$ (for which exists dense sequencing of width $\tilde{h}$), then there is a polynomial algorithm for an arbitrary graph for $h = \tilde{h}$.

Proof. Suppose there is some non-empty graph G having n vertices. It is clear that the length of its optimal sequencing l^* is in the range from 1 (if $n \le h'$ and all vertices are isolated) to n (the graph is a chain) inclusive.

We construct a new graph G', which we obtain from the graph G by adding $m = k * \tilde{h} + r$ isolated vertices to it, where $r = \tilde{h} * \left\lceil \frac{n}{\tilde{h}} \right\rceil - n$, k being some factor. This factor is not less than 0 (if it is sufficient to add $r < \tilde{h}$ vertices to G to obtain dense sequencing, that is, for graphs for which there are optimal sequencings where there are r free positions). It reaches its greatest value in the case of a graph-chain when

we have $n*\left(\widetilde{h}-1\right)=k*\widetilde{h}+r=k*\widetilde{h}+\widetilde{h}*\left\lceil\frac{n}{\widetilde{h}}\right\rceil-n\Rightarrow k=n-\left\lceil\frac{n}{\widetilde{h}}\right\rceil$. Therefore $k\in\overline{0,n-\left\lceil\frac{n}{\widetilde{h}}\right\rceil}$. Now let us construct a sequencing S of G' for $h=\widetilde{h}$ using Algorithm A. It is known that, if there is a dense sequencing for G', it will be optimal and can be obtained in polynomial time.

It is clear that if $m=l^*\ast\widetilde{h}-n$, then S will be dense and its length will be l^*. On the one hand, it cannot be less than l^* since at least l^* places are required to place the vertices of graph G. On the other hand, it cannot be larger than l^*, since in the optimal sequencing S^* for G there are $l^*\ast\widetilde{h}-n$ positions that can be occupied by isolated vertices, and algorithm A finds the optimal dense sequencing, if any. Then if we remove from S all the added isolated vertices, then we obtain the optimal sequencing of S^* for the original problem in polynomial time, since the addition and removal of isolated vertices can be done in linear time.

Similarly to Proof of Theorem 1, if we can find the number of isolated vertices $m=l^*\ast\widetilde{h}-n$ in polynomial time, or, what is the same, the value of the factor k for which the corresponding value is reached, then we can use algorithm A to find the optimal sequencing for any graph for $h=\widetilde{h}$.

Note that $m=l^*\ast\widetilde{h}-n=k*\widetilde{h}+r$ corresponds to the minimum value of k, for which the obtained sequencing S by algorithm A for the graph G' and $h=\widetilde{h}$ will be dense. Indeed, for smaller values of k, obtained sequencing won't be dense, since not enough isolated vertices have been added. For larger values of k, we also obtain dense sequencing, since we add the number of vertices that is a multiple of $\widetilde{h}$.

We apply binary search to this problem. We choose some integer value of $k_0\in\overline{a_0,b_0}=\overline{0,n-\left\lceil\frac{n}{\widetilde{h}}\right\rceil}$, which splits this interval approximately in half. We construct a graph G', for that value of k, then we find the sequencing of S using algorithm A for $h=\widetilde{h}$. If S is dense, then we choose, similarly, $k_1\in\overline{a_1,b_1}=\overline{0,k_0}$, and continue search within this interval. If S has free positions, then we choose $k_1\in\overline{a_1,b_1}=\overline{k_0+1,n-\left\lceil\frac{n}{\widetilde{h}}\right\rceil}$ and similarly continue the search. The algorithm convergence follows from the convergence of binary search for monotonic functions. The search requires logarithmic time, the check can be performed in linear time, so we find the required m in polynomial time.

Since all the steps of the described algorithm can be performed in polynomial time, algorithm A has polynomial complexity under the condition of the theorem, thus we obtained an algorithm that is exact and has polynomial complexity for any graph G and $h=\widetilde{h}$. ■

Corollary 1. If there is an exact algorithm A of polynomial complexity for some fixed sequencing width $\tilde{h}$ and graphs from $D_{\tilde{h}}$ (for which exists dense sequencing of width $\tilde{h}$), then there is a polynomial algorithm for arbitrary graph and for any $h' < \tilde{h}$.

Proof. It follows directly from Theorems 1 and 2. ■

This corollary allows us to generalize the algorithms obtained for the class of graphs for which there are dense sequencings, for the case of all graphs and smaller h. In particular, it allows us to apply algorithms obtained for even values of h to smaller odd ones.

5 Generalization of the algorithm based on the maximum matching

This section will discuss the theorems that prove the feasibility of using and developing modifications of the algorithm based on maximal matching to graphs for which there are dense sequencings.

Further research will use the notion of "undirected graph clique" in the classical sense.

Definition. A clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent [11].

Theorem 1. From the existence of dense sequencing for some graph G for sequencing width $h = \tilde{h}$ follows the existence of disjoint cliques of size $\tilde{h}$ in reachability graph, covering all its vertices.

Proof. Suppose that we have a graph G for which the optimal sequencing S^* of length l^* for $h = \tilde{h}$ is dense.

For any place $i \in \overline{1,l^*}$, since the vertices of $S^*[i]$ are in the same place, by the definition of parallel sequencing, it follows that for any two vertices $v_1 \in S^*[i], v_2 \in S^*[i]$ there is no directed path neither from v_1 to v_2, nor from v_2 to v_1.

It is known that in the reachability graph $\overline{G}$ corresponding to graph G, two vertices v_1 and v_2 are joined by an edge only when there is no directed path neither from v_1 to v_2 nor from v_2 to v_1. From the above, it follows that there are edges in $\overline{G}$ between all pairs of vertices $v_1, v_2 \in S^*[i]$, and hence they form a clique of size $\tilde{h}$ in it.

Since the above reasoning is true for all places in the sequencing S^* and all of them has $\tilde{h}$ vertices of graph G, and that sequencing contains all vertices of graph G, this leads to the conclusion of the statement. ■

The above statement is a necessary condition for the existence of dense sequencing for the graph. Note that in the case of $h = 2$, it is sufficient. For $h > 2$ there are graphs for which there are cliques of size h covering all of its vertices, but dense sequencing doesn't exist.

It is known that pairs of vertices are considered in the algorithm based on the maximum matching, so it is more convenient to apply it to even values of $\tilde{h}$. The following statement proves the feasibility of using an algorithm based on maximal matching for this case.

Theorem 2. Among the maximal matchings M of the reachability graph $\overline{G}$ corresponding to the graph G, there are those that give the optimal sequencing when applying the algorithm based on the maximum matching, when the optimal sequencing for an even width $h = \tilde{h}$ is dense.

Proof. Suppose that a graph G is given for which the optimal sequencing S^* of length l^* for $h = \tilde{h}$ is dense.

We construct for it a reachability graph $\overline{G}$. It follows from Theorem 1 that there exists a cover of vertices of the graph $\overline{G}$ by cliques of size $\tilde{h}$. Let's look at an arbitrary clique from this cover and split the vertices that are belonged to it in pairs (we can do this since $\tilde{h}$ is even).

Let us now consider all such pairs obtained from all the cliques forming the cover. These pairs form the maximum matching for the graph $\overline{G}$, since each vertex belongs to only one pair (pairs contain vertices from one clique and cliques are disjoint), in each pair the vertices are adjacent (belong to the same clique), the pairs cover all vertices (pairs cover all vertices of the cliques, and cliques cover all vertices of the graph).

Consider now the algorithm based on the maximum matching, in which we will place as many pairs as possible on each place. If we apply that algorithm, where we use the resulting set of pairs as the maximum matching, then we can get the optimal sequencing by placing pairs from one clique in one place (vertices from one clique correspond to one place in S^*).■

Note that the validity of the statement does not contradict the fact that for the reachability graph there may be maximal matchings, by using which we will not obtain the optimal sequencing. In fact, such graphs and corresponding matchings exist.

The above statements suggest that the algorithm based on maximal matching can, theoretically, find the optimal solution for problems where the sequencing width is even and the graphs have dense sequencing.

We also see that in its classical form, it is only approximate to this class of sequencing problems.

6 Computational experiment

To experimentally determine the accuracy of an algorithm based on the maximum matching, it was implemented according to algorithm discussed in section 3. This algorithm is further referred to as the classical algorithm based on the maximum matching. Preference was given to vertices with smaller labels, rather than random choice. This will not affect the generality of the results since the randomness of the selection is implemented by the randomness of the graph generation and the labeling

of their vertices. Also, to generalize the algorithm for the case of width $h > 2$ step 5 of the algorithm is repeated until free positions are exhausted or until we can't choose the next pair or single vertex.

During the experiment, only graphs were generated for which there are dense sequencings for even values of h. Their advantages include the fact that for such graphs the exact solution is known by generation. Transitive arrows are not removed from graphs because their presence does not affect the algorithm (they do not affect the reachability graph). The labeling of vertices in the graph is random to implement the randomness of the selection of pairs and vertices that do not belong to the pair.

In this and subsequent experiments, the accuracy of the algorithm based on the maximum matching was verified.

All of the experimental conditions are summarized in Table 1.

Table 1. - Values of parameters used in testing

n	h	Number of tests
$[10,20]$	$\{4,6,8,10\}$	10000
$[21,40]$	$\{4,6,8,10\}$	10000
$[41,60]$	$\{4,6,8,10\}$	10000
$[61,100]$	$\{4,6,8,10\}$	10000

The results of evaluating the accuracy of the classical algorithm based on the maximum matching are shown in Table 2. These results and the following contain three parameters: the number of cases when the algorithm based on the maximum matching found the exact solution; the average deviation of the length of the obtained sequencing from the length of the optimal one, and the number of cases in which the resulting sequencing is twice as long as the optimal sequencing. In this one and following tables AMM stands for algorithm based on maximal matching.

Table 2. - Experimental results for the classical algorithm

Classical algorithm		
AMM is accurate	Average deviation of AMM	Obtained sequencing two times longer than optimal
8688	1.160823	9
5718	1.562121	19
2564	2.165008	4
761	3.391384	0

Test results show that it has very low accuracy. The number of times it gives the optimal solution decreases rapidly. The average deviation of the obtained solutions also grows very fast and differs significantly from the one even for small graphs (21-40 vertices). In addition, there are cases where the sequencing obtained is twice as long as the optimal one, which exceeds the maximum accuracy estimate in [12]. Note that this does not contradict the statement, since the algorithm can sometimes leave

empty positions in sequencing when open vertices are still present. Such behavior is a case when they belong to pairs that cannot be placed in the current step.

From this, we can conclude that as the number of vertices in the graph increases, the number of maximal matchings that give suboptimal solutions increases faster than the number of matchings that give optimal ones.

It is known that the resulting matching is strongly depends on the labels of vertices in the graph. One can assume that if we can relabel the vertices so that algorithm is more likely to find matchings that give optimal sequencing, then we can improve its results. It is also known that the vast majority of optimal solutions for graphs are satisfying the level principle. Therefore, it is natural to expect that if we relabel the vertices of the graph according to the level principle, then we can increase the likelihood of getting a matching, which gives an optimal solution. The results of the corresponding experiment are shown in Table 3.

Table 3. - Experimental results for the classical algorithm with the relabeling of the vertices

Classical algorithm with relabeling		
AMM is accurate	Average deviation of AMM	Obtained sequencing two times longer than optimal
9290	1.083099	0
6485	1.317496	0
3078	1.684051	0
943	2.509551	0

From the experiment we observe that such relabeling slows down the decrease of accuracy of the algorithm and significantly slows the growth of the average deviation. There were no cases where sequencing that are twice as long as optimal were obtained, but the hypothetical possibility of their existence remains.

The sequencings that are twice as long as the optimal one are obtained, since the algorithm can leave empty positions because the open vertices are bound in pairs that cannot be placed. This behavior of the algorithm further aggravates the impact of the initial matching. An obvious way to reduce this effect is to allow splitting the pair if it cannot be placed in the current step, but at least one of its vertices is open. The resulting algorithm is later referred to as modified algorithm based on the maximum matching. The results of the accuracy evaluation of this algorithm are shown in Table 4.

Table 4. - Experimental results for the modified algorithm

Modified algorithm		
AMM is accurate	Average deviation of AMM	Obtained sequencing two times longer than optimal
9031	1.008256	0
8122	1.06869	0
7387	1.166475	0
6418	1.341709	0

From the results we see that such modification significantly improves the accuracy of the algorithm. The number of cases where the algorithm was accurate for graphs with 61-100 vertices increased almost 8.5 times, for other cases this number also increased significantly. The maximum average deviation from the previous experiment has halved and does not exceed 1.5. As expected there were no cases where the length of the resulting solution was twice the length of the exact one.

Note also that despite the smaller average deviation for the case of graphs with 10-20 vertices compared to the previous experiment, the number of cases where the algorithm is accurate is smaller. Therefore, such modification does not always effectively comply with the level principle. Based on the previous results, it is hypothesized that if we apply the modified algorithm to previously relabeled vertices according to the level principle, we will increase accuracy even further. The results obtained are shown in Table 5.

Table 5. - Results of the experiment for the modified algorithm with relabeling of the vertices

Modified algorithm with relabeling		
AMM is accurate	Average deviation of AMM	Obtained sequencing two times longer than optimal
9675	1.006154	0
8913	1.022079	0
8134	1.107181	0
7491	1.229175	0

We see that a significant improvement in performance of the algorithm is achieved. The number of cases where the algorithm was accurate for graphs with 61-100 vertices increased almost 10 times, compared to the classical algorithm. The average deviation for the first three cases is smaller than the average deviation for the first case for the classical algorithm and for the fourth case is comparable to it.

All this indicates the feasibility of introducing modifications to the algorithm. Therefore, the effect of the initial matching on the result obtained is much smaller. Note that the algorithm did not lose the ability to violate the level principle.

It can be assumed that if we apply, in addition to algorithms based on the level principle, such a modified algorithm, we can further improve the overall results, first of all due to cases in which to obtain optimal sequencing it is necessary to violate the level principle. To verify the latter assumption, an experiment was performed comparing the accuracy of a modified maximum matching algorithm with the best of results of lexicographic and double-label algorithms. The latter further along referred to as combined algorithm (CA). The same parameters were used as for the previous experiments.

The results consist of two parts: comparison of the algorithms with each other and comparison of the accuracy of the algorithms by the values of the objective function. A comparison of the algorithms with each other contains the number of test cases in which sequencing with a shorter length is obtained by the AMM; in which a sequencing with a shorter length is obtained by the combined algorithm, and in which sequencings of equal length are obtained by the algorithms. A comparison of the accu-

racy of the algorithms contains the number of cases when an exact solution was obtained by the algorithm based on the maximum matching; when an exact solution is obtained by the CA; the number of cases when one of the algorithms was accurate and the average deviation of the length of the solutions obtained by the algorithms from the length of the optimal one.

The results of the experiment are shown in Table 6.

Table 6. - Results of comparison of accuracy of the modified relabeling algorithm and the combined algorithm

Modified algorithm with relabeling and combined algorithm				
Sequencing obtained by AMM is shorter		Obtained sequencings have the same length		Sequencing obtained by CA is shorter
5		9673		322
30		8954		1016
50		8275		1675
73		7547		2380
AMM is accurate	CA is accurate	One of them is accurate	Average deviation of AMM	Average deviation of CA
9675	9992	9997	1.0230	1
8923	9906	9936	1.02507	1
8206	9815	9865	1.098105	1
7436	9674	9747	1.232059	1

The results obtained confirm that, even with all modifications, the algorithm based on maximal matching is significantly inferior to the algorithms based on the level principle on all indicators. However, we see that the number of cases where the algorithm based on maximal matching finds shorter sequencing increases. In addition, in all these cases, these sequencings were accurate (this can be seen by summing up the number of AMM wins and the number of cases where the CA was accurate and comparing it to the number of cases when one of the algorithms is accurate).

On the other hand, the number of cases when the combined algorithm gave shorter but not accurate sequencing increases with the number of vertices. This may indicate that there is some "barrier" that prevents the combined algorithm from finding the optimal solutions, and the more vertices there are in the graph the more noticeable it is. This barrier may be a level principle, more precisely the necessity to violate it.

Let's return to the idea that the labeling of vertices in the graph strongly influences the result of the algorithm. The experiments with relabeling, according to the level principle, showed the effectiveness of this approach. It can be assumed that, if we randomly relabel the vertices, we will get a new matching and a new resultant sequencing, which may be shorter than the original one. The more times we repeat this procedure, the more likely it is that one of the resulting sequencings will be optimal.

To test this hypothesis, an experiment was conducted in which the algorithm based on the maximum matching was applied to the generated graph, and if it gave a suboptimal solution, then the vertices of the graph were randomly relabeled and the algorithm was applied again. The described steps are repeated until the algorithm finds the

optimal sequencing or until the maximum number of repetitions is exhausted. We call this modification a random algorithm based on maximal matching. In graphs generation the same parameter values as in the previous experiments were used; the maximum number of repetitions for all cases is set to 10.

The results of checking the accuracy of a random classical algorithm based on the maximum matching are given in Table 7. These results contain the same columns as the previous experiments.

Table 7. - Experimental results for the random classical algorithm

Random classical algorithm		
AMM is accurate	Average deviation of AMM	Obtained sequencing two times longer than optimal
9946	1	0
9277	1.034578	0
7088	1.205357	0
3543	1.810129	0

From the results of the experiment we can see that the hypothesis finds its confirmation. Thus, the number of cases where the algorithm was accurate increased significantly, in particular for graphs with 10-20 vertices got almost perfect accuracy, and for the case of graphs with 61-100 vertices the accuracy increased almost 5 times compared to the classical algorithm. The average deviation significantly decreased. There were no cases where the length of the resulting solution exceeds twice the length of the exact one, which may be due to the fact that the matchings that give rise to them are very rare.

A similar experiment was performed for the modified algorithm. The results obtained are summarized in Table 8.

Table 8. - Experiment results for the random modified algorithm

Random modified algorithm		
AMM is accurate	Average deviation of AMM	Obtained sequencing two times longer than optimal
9950	1	0
9834	1	0
9600	1.005	0
9145	1.037427	0

In the experiment obtained results similar to the previous ones. The number of cases where the modified algorithm was accurate exceeds 91% for all graph sizes. For graphs with the number of vertices from 61 to 100 it is 12 times better than the classical algorithm. Similarly, the average deviation decreased.

The results obtained in the last experiment are comparable to those of the algorithms based on the level principle, which confirms the effectiveness of the proposed approach, especially considering that the chosen maximum number of repetitions was relatively small. Experiments with the relabeling of vertices, according to the level

principle, are not listed, because it does not qualitatively affect the operation of the algorithm and therefore its accuracy.

Considering the improvement obtained, the last algorithm was compared with the combined algorithm. The results can be seen in Table 9.

Table 9. - Accuracy comparison results for the random modified algorithm and the combined algorithm

Random modified algorithm and combined algorithm				
Sequencing obtained by AMM is shorter		Obtained sequencings have the same length		Sequencing obtained by CA is shorter
4		9957		39
59		9795		146
132		9518		350
195		9085		720
AMM is accurate	CA is accurate	One of them is accurate	Average deviation of AMM	Average deviation of CA
9960	9995	9999	1	1
9841	9928	9987	1	1
9599	9816	9948	1.014963	1
9160	9674	9867	1.041667	1.009202

From the results we observe that the scores of the random modified algorithm are much higher than the scores of the modified one. The number of cases when it and the combined one give the sequencing of same length exceeds 90%. The number of cases where the sequencing obtained by algorithm has a smaller length is also much larger and increases rapidly with the number of vertices in the graph. In contrast to the previous comparison, there are cases, when the algorithm found shorter sequencings, but they were not optimal. The number of cases in which at least one of the algorithms is accurate for graphs of all sizes exceeds 98.6%, which significantly improves the results when using only algorithms based on the level principle. From this we can conclude that the resulting algorithm not only effectively comply with the level principle, but also violates it if necessary.

All previous results further confirm the effectiveness of random algorithms based on maximal matching, as well as the importance and prospectivity of the modifications of the classical algorithm as such that may violate the level principle.

The disadvantages of the random algorithm compared to the algorithm with relabeling, according to the level principle, include its non-determinism, which greatly complicates the analytical study of its properties.

The results obtained in this section have practically proved the feasibility of using the algorithm based on maximal matching and its modifications to the problems of optimal sequencing with graphs for which there are dense sequencings and even width.

7 Conclusion

The well-known class of discrete optimization problems (which are formulated as optimization problems on graphs), and requiring the development of new effective approximate polynomial algorithms, was investigated.

The theorems allowing to reduce the problem of optimal sequencing with arbitrary graph and sequencing width to the problem with graph, for which exists dense sequencing, and even width, have been proved. It was substantiated that to solve problems from this class it is suitable to apply an algorithm based on maximum matching.

For the algorithm based on maximum matching, several modifications have been developed. Experimental verification has established that they significantly increase its accuracy. In addition, the joint application of the proposed algorithms and known algorithms based on the level principle resulted in an accuracy exceeding 98%.

8 References

1. Garey, M., Johnson, D.: Computers and intractability. W.H. Freeman and Company, New York (2009).
2. Prot, D., Bellenguez-Morineau, O.: A survey on how the structure of precedence constraints may change the complexity class of scheduling problems. Journal of Scheduling. 21, 3-16 (2017).
3. Tanaev, V.S., Kovalyov, M.Y., Shafransky, Y.M.: Scheduling theory. Group Technologies, Minsk, Institute of Technical Cybernetics NAN of Belarus (1998). (in Russian)
4. Pinedo, M.: Scheduling. Theory, Algorithms, and Systems. (2016).
5. Brucker, P.: Scheduling algorithms. Springer, Berlin (2007).
6. Hu, T.: Parallel Sequencing and Assembly Line Problems. Operations Research. 9, 841-848 (1961).
7. Coffman, E., Graham, R.: Optimal scheduling for two-processor systems. Acta Informatica. 1, 200-213 (1972).
8. Garey, M., Johnson, D., Tarjan, R., Yannakakis, M.: Scheduling Opposing Forests. SIAM Journal on Algebraic Discrete Methods. 4, 72-93 (1983).
9. Dolev, D., Warmuth, M.: Scheduling precedence graphs of bounded height. Journal of Algorithms. 5, 48-59 (1984).
10. Fujii, M., Kasami, T., Ninomiya, K.: Optimal Sequencing of Two Equivalent Processors. SIAM Journal on Applied Mathematics. 17, 784-789 (1969).
11. Luce, R., Perry, A.: A method of matrix analysis of group structure. Psychometrika. 14, 95-116 (1949).
12. Bruno, J., Coffman, E.: Computer and job-shop scheduling theory. Wiley, New York (1976).