Розглянуто задачу складання розкладу проходження процедур пацієнтами санаторію. Обрана задача зведена до розширеної задачі пошуку максимального паросполучення в дводольному графі. Наведено доказ NP-повноти сформульованої задачі. Розроблено оптимальний алгоритм її вирішення. Наведений алгоритм реалізований як частина системи управління лікувальним процесом. Проведено порівняльний експеримент наведеного алгоритму з методом повного перебору

Ключові слова: задача про паросполучення, NP-повнота, дводольний граф,оптимальний алгоритм, метод гілок та меж,метод повного перебору

Рассмотрена задача составления расписания прохождения процедур пациентами санатория. Выбранная задача сведена к расширенной задаче поиска максимального паросочетания в двудольном графе. Для сформулированной задачи доказана NP-полнота. Разработан оптимальный алгоритм ее решения. Приведенный алгоритм реализован как часть системы управления лечебным процессом. Проведен сравнительный эксперимент оптимального алгоритма с методом полного перебора.

Ключевые слова: задача о паросочетании, NP-полнота, двудольный граф, оптимальный алгоритм, метод ветвей и границ, метод полного перебора

UDC 519.161
DOI: 10.15587/1729-4061.2017.92226

DEVELOPMENT OF EFFECTIVE ALGORITHM TO FIND AN OPTIMAL SOLUTION TO THE PROBLEM ON GRAPH MATCHING WITH "DISAPPEARING" ARCS

A. Danylchenko
Senior Lecturer*
E-mail: kissann@ukr.net
V. Skachkov
Senior Lecturer**
E-mail: was@ztu.edu.ua
A. Panishev
Doctor of Technical Sciences, Professor, Head of Department**
E-mail: pzs.ztu@gmail.com
*Department of computer engineering***
Department of Software Systems*
***Zhytomyr state technological university
Chernyahivskoho str., 103, Zhytomyr, Ukraine, 10005

1. Introduction

Assigning the procedures in contemporary sanatorium and therapeutic establishments is a complex process, which should consider a sufficiently large number of factors, the main of which are [1]:

– a list of procedures prescribed by doctor;

– operation time of the treatment room;

– throughput capacity of the treatment room (several patients can take one procedure simultaneously);

– duration of the procedure (for different procedures, duration of one procedure is different);

– duration of time of technical break between procedures;

– compatibility of procedures (a patient cannot simultaneously take several procedures).

In addition, the timetable is subject to additional constraint – a patient can take the next procedure after a certain time after taking the previous one. For each pair of procedures, the value of compatibility time can be different).

It is clear that the situation indicated may be extended to a large number of related tasks on making up optimum schedules at the existence of a certain set of constraints. Such tasks include the following problems: allocation of limited resources over time, assignment of the fulfillment of different types of work (operations, tasks, processes).

It is known that effectiveness of process control is connected, first of all, with the degree of automation of administrative decision making support. That is, efficiency of process management improves with the use of problem-oriented tools for administrative decision-making.

Thus, a scientific and practical task of further development and improvement of models and methods for solving the problems on scheduling theory, assigned on transpositions, and devising, based on their principle, effective accurate algorithms and software provision for the solution of tasks on efficient optimization of therapeutic process is absolutely relevant.

2. Literature review and problem statement

A discrete nature of operations to assign the procedures in sanatorium and therapeutic establishments, as well as the limited possibilities of their conducting that are manifested, for example, in the assigned duration for conducting a procedure with one patient, mutual compatibility of procedures, maximum throughput of treatment rooms, etc., require coherence of operations for assigning the procedures

in space and over time [1]. Such problems are successfully solved by mathematical apparatus of scheduling theory – an area of applied mathematics that studies the models of arranging the operations works methods for making up schedules [2], allocation of limited resources over time, assigning the execution of different kinds of work (operations, tasks, processes).

The problems on making up a schedule can be interpreted as tasks on the allocation of resources (resources in a broad sense are understood as material, temporal resources, etc.) with the assigned limitations (by the amount of resources, time for their creation, processing, etc.) [3]. Sources [4] describe a task on making up a schedule for teaching classes at school, institute of higher education), etc.

A problem of making up a schedule for taking the procedures by patients of a sanatorium, in contrast to the task of finding maximum graph matching in the bipartite graph, contains a constraint that prevents the application, for the time being, of precise effective algorithms for its solution. This limitation is in the prohibition, in the selection of one arc of a bipartite graph, to include other arcs in the solution, creating analogy with the disappearing arcs. [1, 5] give a substantive and formal statement of the problem on making up a schedule for taking the procedures by patients of a sanatorium. It was demonstrated that the problem is reduced to the extended task of searching for maximum graph matching in a bipartite graph. Authors of present article are not aware of other papers, except for [1], which exlored the problems on graph matching with disappearing arcs.

The problems on making up an optimum schedule are frequently solved using modifications of the classical problem on graph matching [6], different statements of which are given in [7]. This problem is to find in the arbitrary weighted graph a maximum graph matching at minimum cost, and in [8] it is solved over time O(nlog(n)), where n is the order of input matrix.

Solving the problems on scheduling theory presents a known complexity. By the content, these problems are related to the class of combinatory ones, which are solved by using a well-known method of branches and boundaries. It is commonly applied for such NP-complete problems as the travelling salesman problem [9] and the knapsack problem.

A solution of the problem on finding maximum graph matching in a bipartite graph is given and is examined in many sources. But there is no statement for the problem on graph matching with disappearing arcs at all. Consequently, the proof of NP-completeness was not examined by authors either.

In article [6], authors calculate complexity and approximated results of the solution to the problem on finding maximum graph matching in the modified problem on graph matchings and demonstrate that this problem is NP-complete.

Many publications proposed algorithms for solving the classical problem on graph matching, which made it possible to reduce computational complexity of the program realizations [10, 11].

In [12], authors developed an algorithm for solving the problem on the planner of tasks by the modification of Hungarian algorithm.

[13] describes a new method to solve the problem about weighted graph matching whose maximum graph matching is found over time $O(n^3)$. In future, it is planned to modify the problem on graph matching with "disappearing" arcs, by adding weight to arcs.

3. The aim and tasks of the study

The purpose of present work is to analyze the problem on making up a schedule for taking the procedures by patients of a sanatorium and to develop an optimal algorithm for its solution.

To achieve the set aim, the following tasks were to be solved:

– an analysis of the problems on effective organization of therapeutic process, a systematization of theoretical and practical results, obtained in the area of solving the tasks of present study;

– construction of a mathematical model for the problem on graph matching with "disappearing arcs" and proof of NP-completeness;

– development of efficient search algorithm for optimal solutions;

– conducting a computational experiment and analysis of the obtained results.

4. Materials and methods of research

A number of problems about graph matchings in bipartite graphs contain a constraint, which forbids, in the selection of one arc, to include in the solution another arc. For example, if arc (x, y) is already contained in the graph matching, then arc (v, w) cannot belong there. Such arcs are called incompatible. They are included in the conditions for the problem on graph matching with disappearing arcs (PGDA). It is obvious that PGDA is the assignment problem with the requirements for incompatibility of some pairs of operations and their performers.

Let us examine one of the applied versions of PGDA – a problem on the distribution in time of health-improving procedures among the patients in sanatorium.

A sanatorium provides a list of different procedures. A patient must take a therapeutic course of all or several procedures of this list. For each particular procedure, a schedule for its conducting is set in the form of a sequence of time gaps. The patient is assigned with taking not more than one procedure on the list. It is required to find a correspondence between a set of all procedures and the set of all time gaps:

a) that prevents assigning a procedure to several patients;

b) that provides for maximum period of taking all procedures.

For example, it is necessary to allocate procedures No. 1 and No. 2 to two patients in sanatorium. The first patient is to take both procedures, the second one is to take only procedure No. 2. The operating schedules for treatment rooms No. 1 and No. 2 are assigned (Fig. 1).

Let us describe conditions of the stated problem in terms of bipartite graphs. Let G=(X,Y,E) is the bipartite directed graph, where X is the set of apexes x_i, each of which corresponds to the possible interval of taking the procedures, i=1,m; Y is the set of apexes y, j=1,n that correspond to the set of all procedures, assigned to the patients. Arc $(x_i,y_j)\in E$ when, and only when, procedure y_j can be taken in time interval x_i.

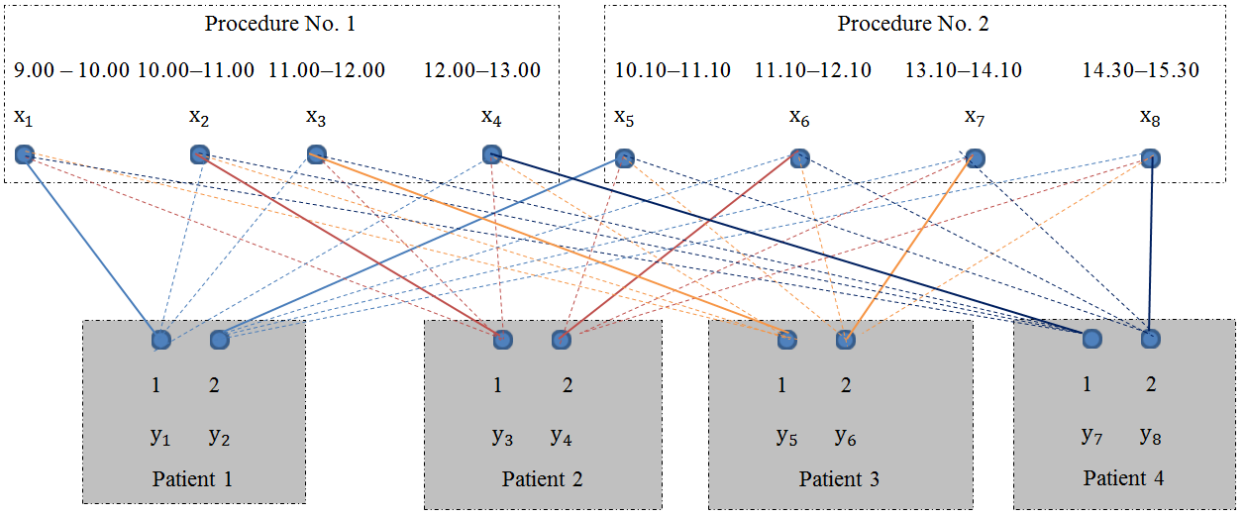

Fig. 1. Example of solving a problem on assigning the procedures

It is obviously that the solution of the problem is a maximum graph matching in a bipartite graph [1]. In Fig. 1, its edges are shown in the thickened lines.

Let us add to the conditions of the given example the following limitation: the second procedure is assigned after the first one to happen not earlier than in an hour. Then the graph matching, which includes arc (x_1, y_1), must be automatically left by arc (x_5, y_2) that is not compatible with arc (x_1, y_1). In a general case, the condition of incompatibility is assigned on a certain subset of arcs of a bipartite directed graph. It is required to find a graph matching, which includes the maximum number of compatible arcs.

In the process of stepwise construction of such graph matching, some arcs will disappear, thus changing the structure of a bipartite directed graph. The examined GDA is a problem on maximum graph matching in a bipartite directed graph with the assigned subset of arcs (x_i, y_j). For each arc (x_i, y_j) of the directed graph, we determined the subset of arcs $C_{i,j}$, not included in the feasible solution if the arc (x_i, y_j) is included in it. The arcs of set $C_{i,j}$ disappear from graph G at the inclusion of arc (x_i, y_j) and they become visible again at its exclusion.

The relation of incompatibility of arc (x_i,y_j) with the subset of arcs $C_{i,j}$ will be denoted as:

$$(x_i,y_j)\to C_{ij}=\left\{\left(x_{i_1},y_{j_1}\right),\left(x_{i_2},y_{j_2}\right),\ldots,\left(x_{i_k},y_{j_k}\right)\right\}. \qquad (1)$$

We register that arc (x_i, y_j) cannot be included in the graph matching together with any arc,

$$\left(x_{i_p},y_{j_p}\right)\in C_{ij},p\in\{1,2,\ldots,k\},k=\left|C_{ij}\right|.$$

Let us note that relation (1) is considered to be not assigned on subset $C_{i,j}$.

Let us estimate the time cost of PGDA. For this purpose, we shall formulate it in the terms of properties recognition, which include the condition and the question.

Condition. Bipartite directed graph is assigned:

$$G=(X,Y,E),$$

where X, Y is the set of apexes,

$$|x|=m,|Y|=n,E,$$

a set of arcs:

$$\left(x_i,y_j\right),\ x_i\in X,\ y_j\in Y.$$

Each arc (x_i, y_j) is found from the relation of incompatibility with any arc of subset $C_{i,j}$, $\varnothing C_{i,j}E$.

Question. Is there a graph matching from the compatible arcs with power min(m, n) in the directed graph G?

Theorem. A problem about graph matching with disappearing arcs (PGDA) NP-complete in the strong sense.

Proof. To prove the theorem, it is necessary to select a certain NP-complete problem and to reduce it to the PGDA problem. Let us take as such a problem the problem "On graph matchings, limited by weight" (GLBW). We shall convert into PGDA a NP-complete in the strong sense problem on graph matching, limited by weight (GLBW).

Let us state the GLBW problem in the terms of properties recognition.

Condition. The disjoint sets X and Y are assigned:

$$|X|=|Y|=m,$$

dimensions:

$$s(a)\in Z^+,$$

of all elements:

$$a\in X\bigcup Y,\ Z^+,$$

a set of positive integer numbers and restriction vector $B_1,\ldots,B_m$ with the non-negative integers, different numbers in pairs.

Question. Is there a partition of set X∪Y into m disjoint subsets $A_1,A_2,\ldots,A_m$, such as each of them contains one element from X and Y and:

$$\sum_{a\in A}S(a)=B_i \text{ for all } i=\overline{1,m}.$$

Let us build, by the GLBW problem, a special case of PGDA. We shall construct a bipartite directed graph with the sets of apexes

V_1 and V_2 , $|V_1| = |V_2| = m$.

We shall assign weights to the apexes of sets V_1 and V_2, equal to the dimensions of elements from sets X and Y, respectively. Two apexes:

$k \in V_1$ and $l \in V_2$,

will be connected by arc (k, l), if there exists such a coordinate of limitations vector B_i as:

$S(k) + S(l) = B_i$; $S(k) = S(a)$, $a \in X$; $S(l) = S(a)$, $a \in Y$.

Let us agree that arcs in the directed graph disappear according to the following rule: if we select an arc with the sum of weights of the initial and final apex, equal to B_i, then all arcs (k, l) are removed, for which:

$S(k)+S(l)=B_i$.

Then, if the GLBW problem has a solution, then it is the solution for particular PGDA problem.

Conversely, assume the PGDA problem has a solution. In the directed graph it is represented by perfect graph matching. For its ith arc:

$(k,l), l = 1,m$,

the sum of weights of apexes $k \in V_1$ and $l \in V_2$ is equal to B_i. Then m pairs of (k, l) apexes and m – sums B_i, among which there are no equals, are the solution for the GLBW problem. Assume that the vector of restrictions contains several sums, equal to B_i. Let us select any arc (k, l) in the directed graph for which:

$S(k) + S(l) = B_i$,

and remove all the remaining arcs with the same sum B_i of weights of the apexes. It is obvious that the obtained directed graph causes a change in the values in the initial vector of restrictions, whence it follows that the GLBW problem does not have a solution.

A reduction in the number of graph matchings, which are analyzed, can be achieved due to taking into account the corollary $C_{i,j}$ in the selection of certain graph matching (x_i, y_j) at each sequential step of the algorithm. Then, when selecting any arc (x_i, y_j) for its inclusion in the set of graph matchings at iteration t, the set of graph edges for further analysis can be represented as:

$$E^{t+1}\Big|_{(x_i,y_j)} = E^t - C_{i,j}.$$

In this case, we will obtain at each iteration a set of lower power E^{t+1} while the region of analysis narrows in comparison with the brute force method.

Algorithm on the base (basis) of the branch and bound method. For the guaranteed finding of the largest graph matching M in the algorithm under development, it is necessary to realize a sequential search for all possible variants of the construction of graph matchings on the assigned graph

$G = (X, Y, E)$,

taking into account assigned limitations C. For this purpose, it is necessary to organize a cycle along all apexes of set Y and inserted cycle along the subset of arcs incident to the current apex y_i.

Satisfying the equality will be a sufficient condition for the optimality of the obtained variant of solution to the problem:

$\|M\| = \|Y\|$.

Upon finding maximum graph matching, execution of the algorithm can be completed.

A necessary condition for the optimality of the obtained variant of solution to the problem is the fulfillment of condition:

$\|M\| = \max$,

that is, finding graph matching M of maximum power. At each step of the algorithm, we shall calculate maximally possible graph matching by the number of arcs (x_i, y_j) with different y.

If, under completion of sequential search for all apexes of set Y, a sufficient condition of optimality is not satisfied, then at the assigned limitations, it is possible to find an optimal solution. The obtained solution will provide for the assignment of the largest possible quantity (but not all assigned) of procedures at the assigned limitations.

To describe the algorithm, based on the branch and bound algorithm, it is necessary to introduce the description of the following objects (B_p, S, M, U, BR, RB).

1. The rule of branching B_p describes the process of branching in the problem about transpositions. The purpose of branching is the partition of the set of complete transpositions into disjoint subsets. These subsets are represented in the form of apexes. Transposition $\pi_y^{M_y}$, is set in a correspondence to each apex, assigned on set M_y for:

$M_y \subseteq N = \{1,2,\ldots,n\}$.

Apex, marked $\pi_y^{M_y}$, is the set of complete transpositions. Partial transposition $\pi_y^{M_y}$ is the ancestor of each transposition, included in set $\{\pi_y^{M_y} \circ\}$. Branching in the apex

$\pi_b^{M_b} = (b_1, b_2, \ldots, b_k)$

is the process of partition of set $\{\pi_b^{M_b} \circ\}$ into two subsets of arcs into one, from which the arc (x_i, y_j) enters, and the second set – excluding this arc. For apex y_k, we shall successively investigate the edges

$(x_i, y_k) \in E$, $i = 1...m$.

Assume that a certain edge is found

$(x_i, y_k) \in E$, $y_k \in P_j$,

candidate for inclusion into M, that is, none of the apexes x_i and y_k is incident to the edges, already included in graph matchings M. We consider the assigned limitations, and for this purpose, we temporarily exclude from the graph

$G = (X, Y, E)$,

arcs $(x_i, y_k) \to C_{i,k}$. That is, at each step of the algorithm, we conduct branching by including the arc

$(x_i, y_k) \in E$

into a graph matching, we obtain one problem (including) and reconstruct graph G=(X,Y,E), excluding from it all arcs by the conditions of corollary $C_{i,j}$, and the second (excluding) problem, where from the graph

$G = (X,Y,E)$

we exclude arc

$(x_i, y_k) \in E.$

Including problem with arcs:

$E^{/}_{i,j} \setminus C_{i,j},$

where E is the set of all arcs.

Excluding problem with arcs:

$E^{2}_{i,j} \setminus C^{zag}_{i,j},$

where $C^{zag}_{i,j}$ is the set of arcs from the condition of corollary $C_{i,j}$ – arc from i to j. Not a single arc is lost. If we combine sets of both problems, then we shall obtain a general problem. That is, one problem without one arc, and the second one – with it, but without the corollary conditions.

The rule of selection S serves for selecting the subsequent apex of branching from the current set of active apexes. Apex π_y is called a current active apex when, and only when, it is generated but is not yet excluded and not exposed to branching. The algorithm starts its operation by selecting $e = \pi^{\varnothing}$ as the first apex of branching. The work of algorithm is completed when the next apex of branching is the complete solution.

The rule of branching for the examined problem is as follows: current active apex π_a is selected at each step of the algorithm. The selected apex must have the largest lower cost estimate $M(\pi_a)$. The lower cost estimate is considered when, and only when, the upper cost estimate U is equal for all active apexes. If upper estimations for the active apexes are not equal, we shall select for the branching an apex with the largest upper estimation and maximum lower estimation. We select apex π_a, in which

$M(\pi_y) = \max(M)$ and $U(\pi_y) = \max(U).$

Thus, at step 1 for the including problem, when graph matching M includes an arc,

$(x_i, y_k) \in E, \ M_1 = \{(x_i, y_k)\}$ and $|M| = 1,$

for the second problem of the algorithm's step 1:

$M = \varnothing$ and $|M| = 0,$

since the graph matching did not include any arc, but we simply excluded from graph G=(X,Y,E) the arc $(x_i,y_k) \in E$.

Function of lower estimate M assigns in a correspondence to each partial solution $\pi_y \in P$ a real number $M(\pi_y)$, which is the lower cost estimate for all complete solutions. For all

$\pi_y, \ \pi_z \in PM,$

has the following attribute: if π_z is the π_y, descendant, then

$M(\pi_z) \geq M(\pi_y).$

For the problem on graph matching with the disappearing arcs, lower estimate is the quantity of arcs, included in the maximum graph matching at each step of branching. Thus, for the first active apex

$M(\pi_1) = 0,$

since not any arc is introduced into the set of solutions as yet. For the including problem, when the graph matching includes one arc but several arcs were excluded (according to the corollary conditions) $U(\pi_2)$ may not equal count(y_i). Similarly for the excluding problem, where arc $(x_i,y_k) \in E$ is excluded from the graph if it is the only arc for apex Y, then:

$U(\pi_3) = \text{count}(y_i) - 1.$

The upper cost estimate U is the cost of a certain complete solution π^N_u, known from the start of fulfilling the algorithm. For the examined particular problem for calculating upper estimation of the first active apex, we count a number of procedures, which should be assigned:

$U(\pi_1) = \text{count}(y_i).$

At further branching, initial choice of the solution whose upper cost estimate is close to the optimum value substantially reduces the total volume of computation.

The optimum algorithm for solving the problem on graph matching with disappearing arcs contains the following steps:

We build a bipartite graph G=(X,Y,E), in which:

– X is the set of apexes of the graph, which correspond to possible time gaps in taking the procedures $\|X\| = m$;

– Y is the set of apexes of the graph, which correspond to the procedures, assigned to the patients in sanatorium, $\|Y\| = n$;

– P is the set of patients;

– E is the set of edges of the graph.

Edge $(x_i, y_k) \in E$, $x_i \in X$, $y_k \in Y$ i=1...m, k=1...n when the procedure of patient y_k may be assigned for the time interval x_i. Assigned relations of corollary $C_{i,j}$. A set of edges, included in the maximum graph matching:

$M = \varnothing$ and $|M| = 0.$

Assume:

$k = q, \ q = 1.$

1. We consecutively iterate through apexes, starting at k=q, apex of the set Y. If all apexes are selected, we shall proceed to point 5.

2. For the selected active apex, we apply the rule of branching B.

3. For each apex, obtained in the process of branching, we calculate the lower and upper cost estimates and use the rule of selection S.

4. We verify existence of the arcs:

$(x_i, y_r) \in E, \ x_i \in X, \ y_r \in Y, \ i = 1...m, \ r = k+1...n,$

that is, those incident to the subsequent apexes in set Y (there are also procedures, for which the time of their taking is not assigned). If each of these apexes has at least one arc, incident to it, then $M=M\cup(x_i,y_k)$. If k=n, proceed to point 2. Otherwise k=k+1 and we shall proceed to point 3.

5. We check that the obtained graph matching contains all the procedures assigned. If $\|M\|=n$, the solution is found, the algorithm is completed. If $\|M\|\neq n$, then proceed to point 2. Otherwise, select from the obtained solutions the graph matchings of maximum power.

In the following chaper there is an example of solving the problem about graph matching with disappearing arcs using the optimum algorithm proposed.

5. Results of examining a solution to the problem on graph matching with "disappearing arcs"

Let us examine example.

Step 1 (Fig. 3, *a*). The constructed bipartite graph G= =(X, Y, E) contains a set of apexes of the graph, which correspond to the possible time gaps in taking the procedures:

X={9.00–10.00, 10.00–11.00, 11.00–12.00, 12.00–13.00, 10.10–11.10, 11.10–12.10, 13.10–14.10, 9.30–10.30, 10.30–11.30, 11.30–12.30, 9.00–10.00, 10.00–11.00, 11.00–12.00},

$$|X|=13,$$

a set of apexes of the graph, which correspond to the procedures, assigned to the patients in sanatorium:

$$Y=\{1_1,1_2,1_3,2_2,2_3,2_4,3_3,3_4\},\ |Y|=8.$$

E is the set of edges of the graph, which correspond to a case when the procedure of patient y_k can be assigned for the time interval x_i:

$$E=\{(x_1,y_1),(x_2,y_1),(x_3,y_1),(x_4,y_1),(x_5,y_2),(x_6,y_2),(x_7,y_2),(x_8,y_3),(x_9,y_3),(x_{10},y_3),(x_5,y_4),(x_6,y_4),(x_7,y_4),(x_8,y_5),(x_9,y_5),(x_{10},y_5),(x_{11},y_6),(x_{12},y_6),(x_{13},y_6),(x_{11},y_7),(x_{12},y_7),(x_{13},y_7),(x_8,y_8),(x_9,y_8),(x_{10},y_8)\}$$

Let us introduce the following limitations.

– Impossibility to assign one and the same procedure to different patients for the same time.

The assigned limitations will be determined by the corollary C set, such as:

$$(x_1,y_1)\to C_{1,1}=\{(x_2,y_1),(x_3,y_1),(x_4,y_1),(x_8,y_3)\};$$

$$(x_2,y_1)\to C_{2,1}=\{(x_1,y_1),(x_3,y_1),(x_4,y_1),(x_8,y_3),(x_9,y_3),(x_5,y_2)\};$$

$$(x_3,y_1)\to C_{3,1}=\{(x_1,y_1),(x_2,y_1),(x_4,y_1),(x_8,y_3),(x_9,y_3),(x_{10},y_3),(x_5,y_2),(x_6,y_2)\};$$

$$(x_4,y_1)\to C_{4,1}=\{(x_1,y_1),(x_2,y_1),(x_3,y_1),(x_8,y_3),(x_9,y_3),(x_{10},y_3),(x_6,y_2)\};$$

$$(x_5,y_2)\to C_{5,2}=\{(x_6,y_2),(x_7,y_2),(x_8,y_3),(x_9,y_3),(x_2,y_1),(x_3,y_1),(x_5,y_4)\};$$

$$(x_6,y_2)\to C_{6,2}=\{(x_5,y_2),(x_7,y_2),(x_3,y_1),(x_4,y_1),(x_9,y_3),(x_{10},y_3),(x_6,y_4)\};$$

$$(x_7,y_2)\to C_{7,2}=\{(x_5,y_2),(x_6,y_2),(x_7,y_4)\};$$

$$(x_8,y_3)\to C_{5,3}=\{(x_9,y_3),(x_{10},y_3),(x_1,y_1),(x_2,y_1),(x_5,y_2),(x_8,y_5),(x_8,y_8)\};$$

$$(x_9,y_3)\to C_{9,3}=\{(x_8,y_3),(x_{10},y_3),(x_2,y_1),(x_3,y_1),(x_5,y_2),(x_6,y_2),(x_9,y_5),(x_9,y_8)\};$$

$$(x_{10},y_3)\to C_{10,3}=\{(x_8,y_3),(x_9,y_3),(x_4,y_1),(x_3,y_1),(x_6,y_2),(x_{10},y_5),(x_{10},y_8)\};$$

$$(x_5,y_4)\to C_{5,4}=\{(x_6,y_4),(x_7,y_4),(x_5,y_2),(x_8,y_5),(x_9,y_5),(x_{12},y_6),(x_{13},y_6)\};$$

$$(x_6,y_4)\to C_{6,4}=\{(x_5,y_4),(x_7,y_4),(x_6,y_2),(x_9,y_5),(x_{10},y_5),(x_{13},y_6)\};$$

$$(x_7,y_4)\to C_{7,4}=\{(x_6,y_4),(x_5,y_4),(x_7,y_2)\};$$

$$(x_8,y_5)\to C_{8,5}=\{(x_9,y_5),(x_{10},y_5),(x_8,y_3),(x_8,y_8),(x_5,y_4),(x_{11},y_6),(x_{12},y_6)\};$$

$$(x_9,y_5)\to C_{9,5}=\{(x_8,y_5),(x_{10},y_5),(x_9,y_3),(x_9,y_8),(x_5,y_4),(x_6,y_4),(x_{12},y_6),(x_{13},y_6)\};$$

$$(x_{10},y_5)\to C_{10,5}=\{(x_8,y_5),(x_9,y_5),(x_{10},y_3),(x_{10},y_8),(x_6,y_2),(x_{13},y_6)\};$$

$$(x_{11},y_6)\to C_{11,6}=\{(x_{12},y_6),(x_{13},y_6),(x_{11},y_7),(x_8,y_5)\};$$

$$(x_{12},y_6)\to C_{12,6}=\{(x_{11},y_6),(x_{13},y_6),(x_{12},y_7),(x_5,y_4),(x_8,y_5),(x_9,y_5)\};$$

$$(x_{13},y_6)\to C_{13,6}=\{(x_{11},y_6),(x_{12},y_6),(x_{13},y_7),(x_5,y_4),(x_6,y_4),(x_9,y_5),(x_{10},y_5)\};$$

$$(x_{11},y_7)\to C_{11,7}=\{(x_{12},y_7),(x_{13},y_7),(x_{11},y_6),(x_8,y_8)\};$$

$$(x_{12},y_7)\to C_{12,7}=$$
$$=\{(x_{11},y_7),(x_{13},y_7),(x_{12},y_6),(x_8,y_8),(x_9,y_8)\};$$

$$(x_{13},y_7)\to C_{13,7}=$$
$$=\{(x_{11},y_7),(x_{12},y_7),(x_{13},y_6),(x_9,y_8),(x_{10},y_8)\};$$

$$(x_8,y_8)\to C_{8,8}=$$
$$=\{(x_9,y_8),(x_{10},y_8),(x_8,y_3),(x_8,y_5),(x_{11},y_7),(x_{12},y_7)\};$$

$$(x_9,y_8)\to C_{9,8}=$$
$$=\{(x_8,y_8),(x_{10},y_8),(x_9,y_5),(x_9,y_3),(x_{12},y_7),(x_{13},y_7)\};$$

$$(x_{10},y_8)\to C_{10,8}=$$
$$=\{(x_9,y_8),(x_8,y_8),(x_{10},y_3),(x_{10},y_5),(x_{13},y_6)\}.$$

A set of edges, included in the maximum graph matching, M=Ø.

A graphic representation of the branching algorithm is shown in Fig. 2

Next, we present a step-by-step process of solving the problem on graph matching with disappearing arcs by the modified branch and bound method (Fig. 3).

Mmax=8 Maximum graph matching is found.

The main goal of comparative computational experiment is the estimation of time cost for the fulfillment of calculations by the optimum algorithm and the brute force method. The experiment was conducted on the computational platforms, which have divergent characteristics (number of cores in processor, clock frequency, memory size, etc.). The result of experiment is the choice of the most effective algorithm for solving the problem on making up a schedule for taking therapeutic procedures by patients in a sanatorium.

A criterion of effectiveness is the minimization of time for performing the calculations in the search for graph matching as it is in the bipartite graph with disappearing arcs.

The experiment was conducted at sanatorium "Denishi" (Ukraine). The experiment employed the authoring program ICS_DENISH, which can solve the problems on making up a schedule for taking the procedures by patients in the sanatorium. To solve the problem, the program applies the brute force method and the optimum algorithm for solving a problem on the search for maximum graph matching in a bipartite graph with disappearing arcs.

Comparative computational experiment was carried out on a series of random conditions of the problem, obtained from actual patients in the sanatorium by the time sample. Characteristics of the computers, which were employed for conducting the experiment, are given in Table 1.

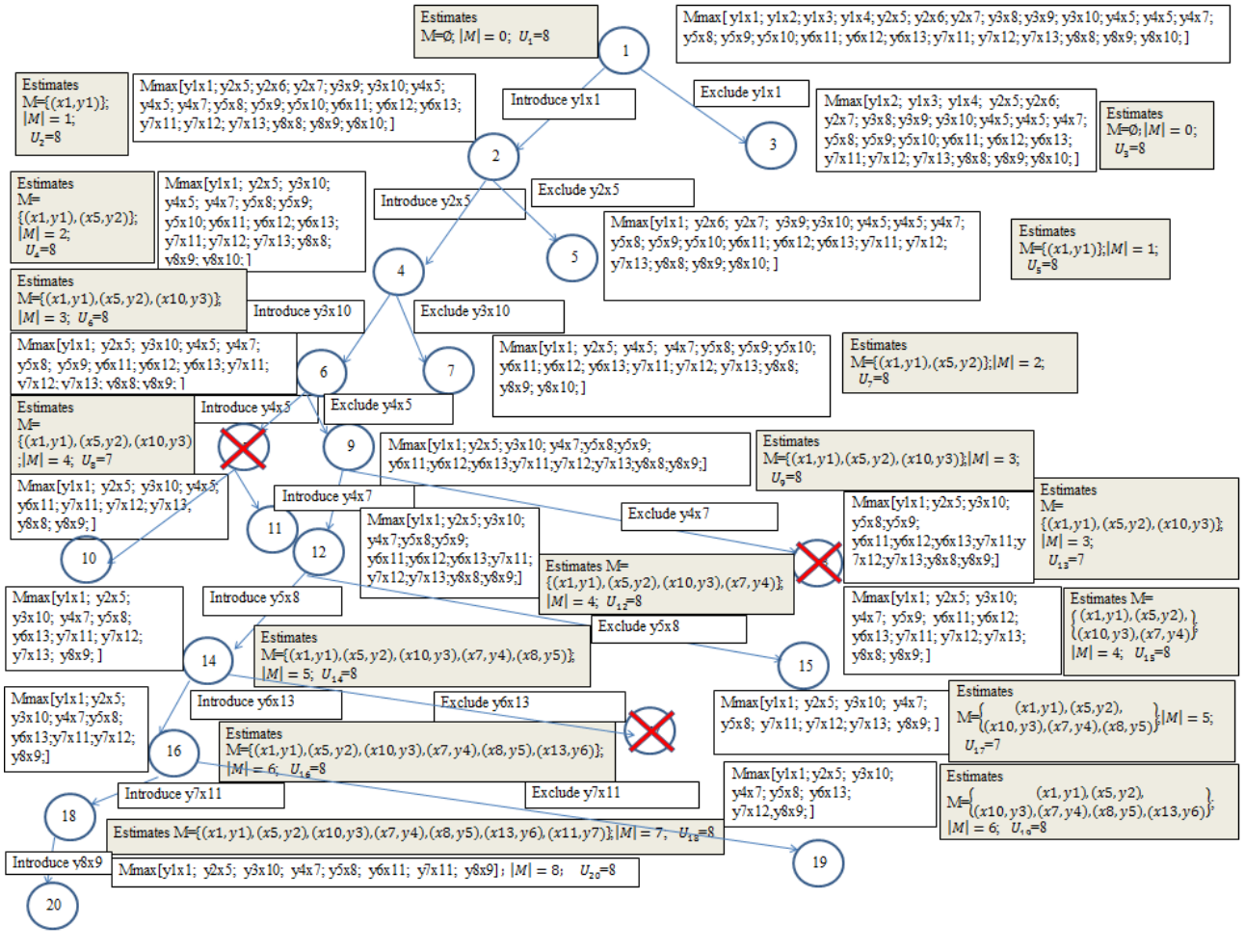

Fig. 2. Branching procedure

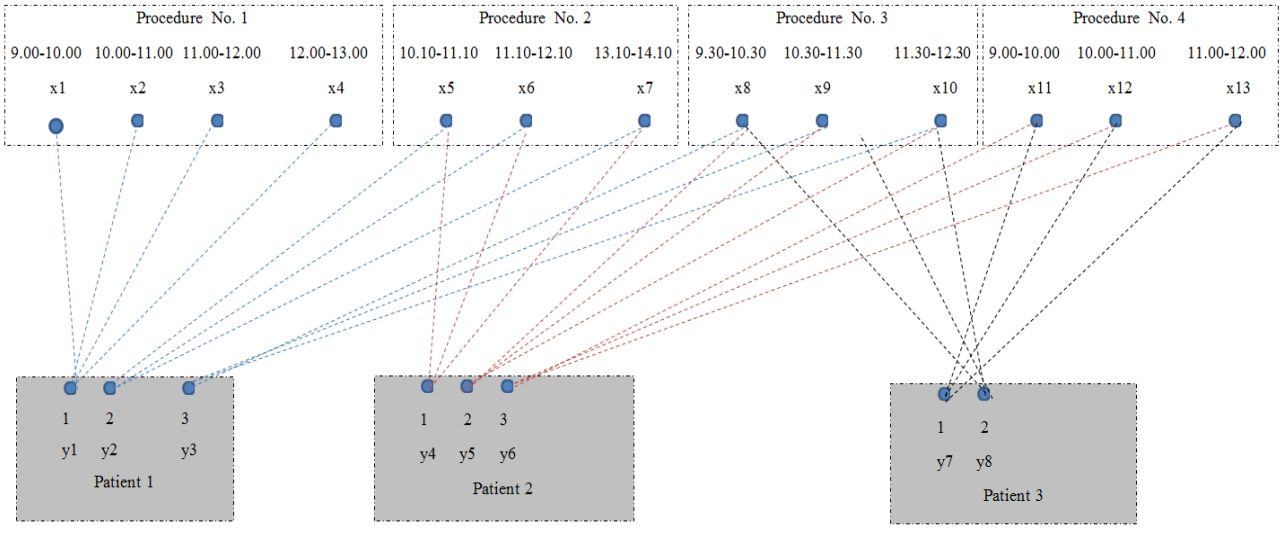

Fig. 3. Initial data for solving the problem on graph matching with disappearing arcs

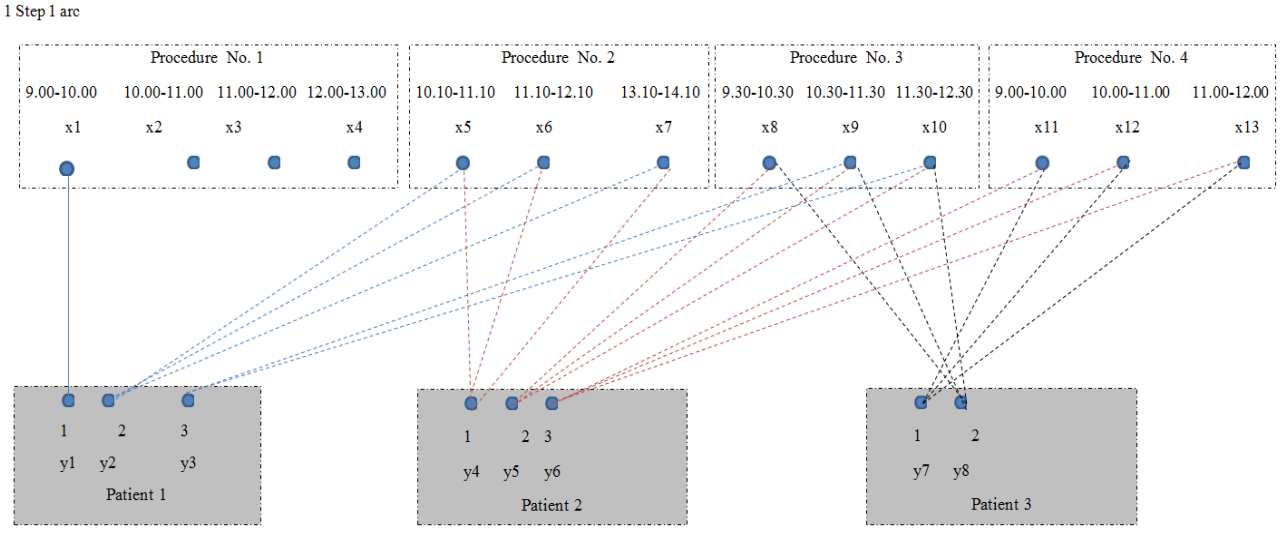

Fig. 4. Selection of arc (x_1,y_1) and appropriate modification of the bipartite graph

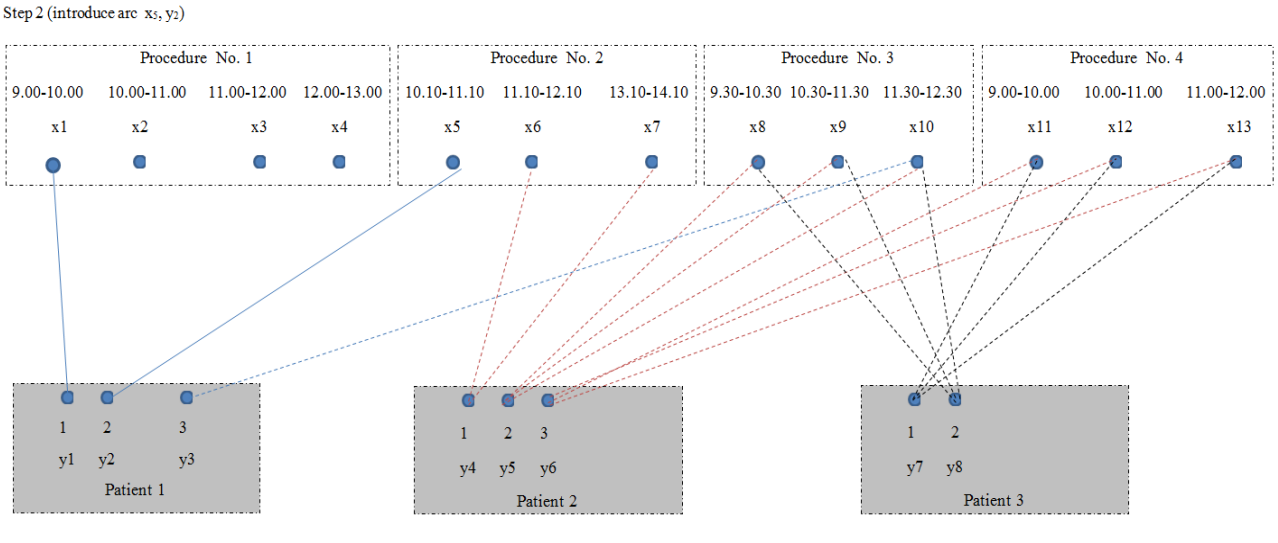

Fig. 5. Selection of arc (x_5,y_2) and appropriate modification of the bipartite graph

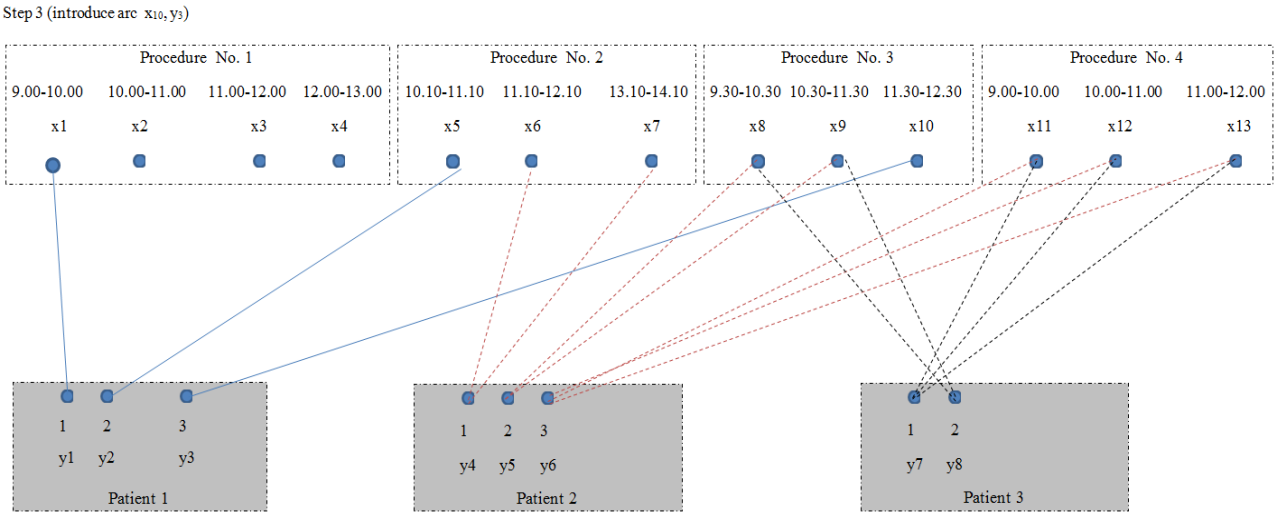

Fig. 6. Selection of arc (x_{10},y_3) and appropriate modification of the bipartite graph

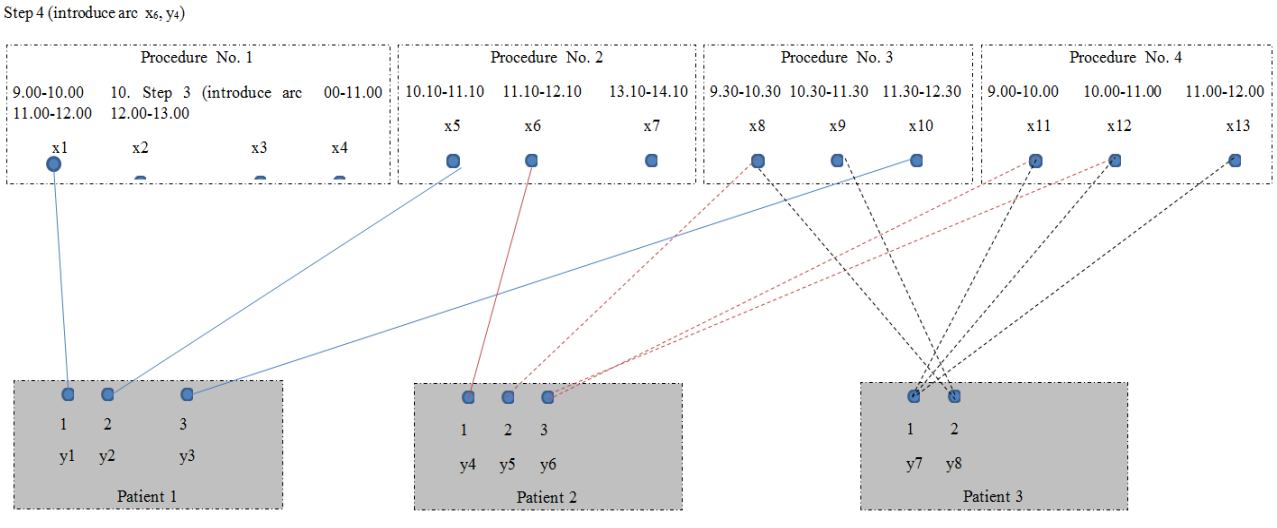

Fig. 7. Selection of arc (x_6,y_4) and appropriate modification of the bipartite graph

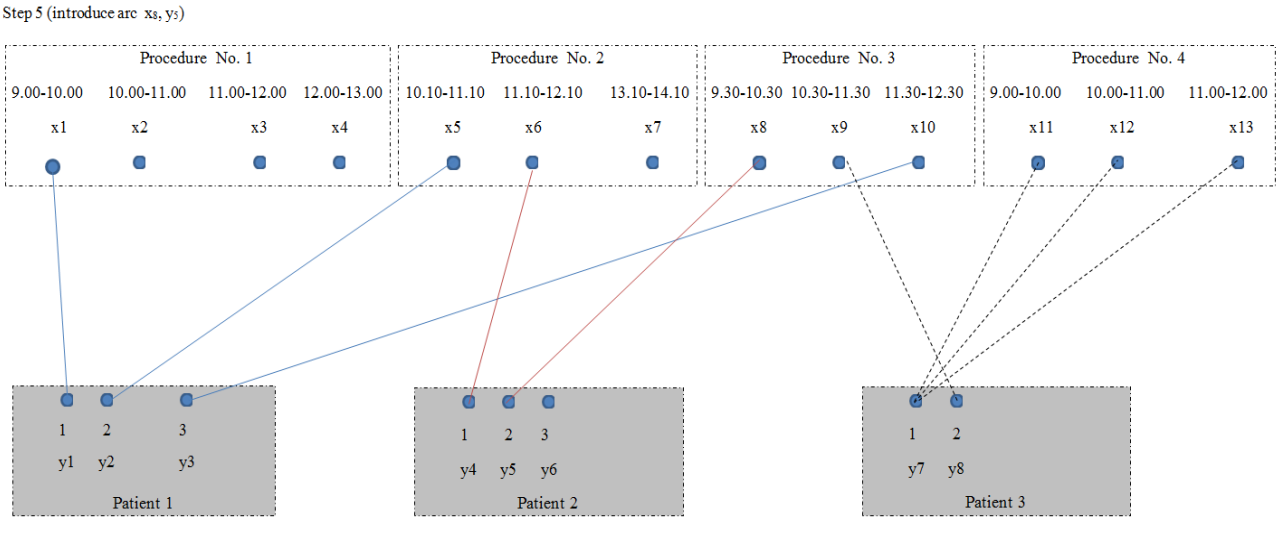

Fig. 8. Selection of arc (x_8,y_5) and appropriate modification of the bipartite graph

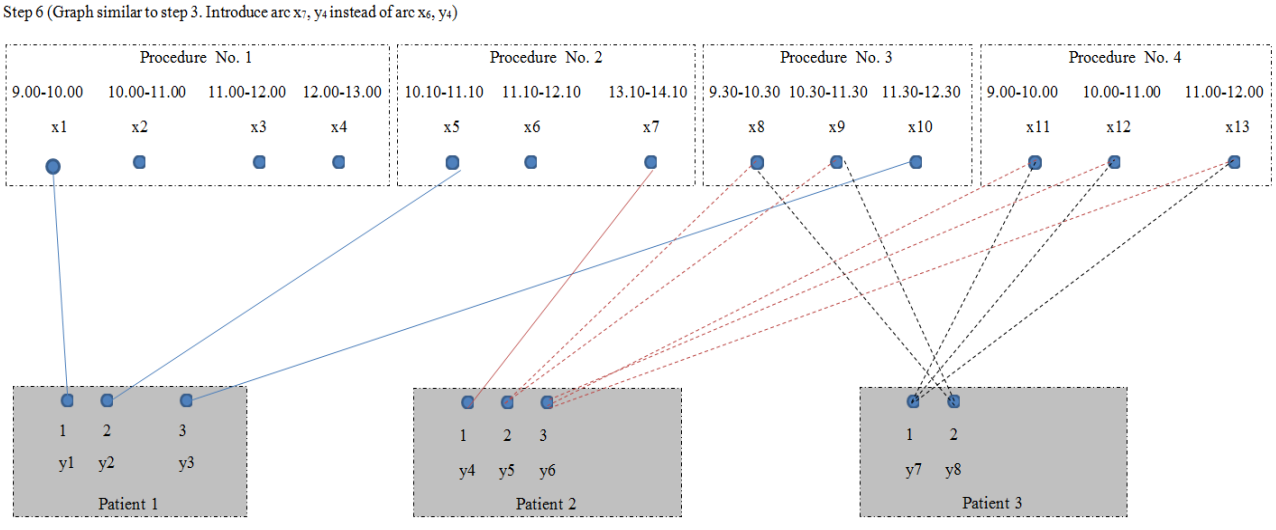

Fig. 9. Removal of arc (x_6,y4) from the solution of the problem and selection of arc (x_7,y_4)

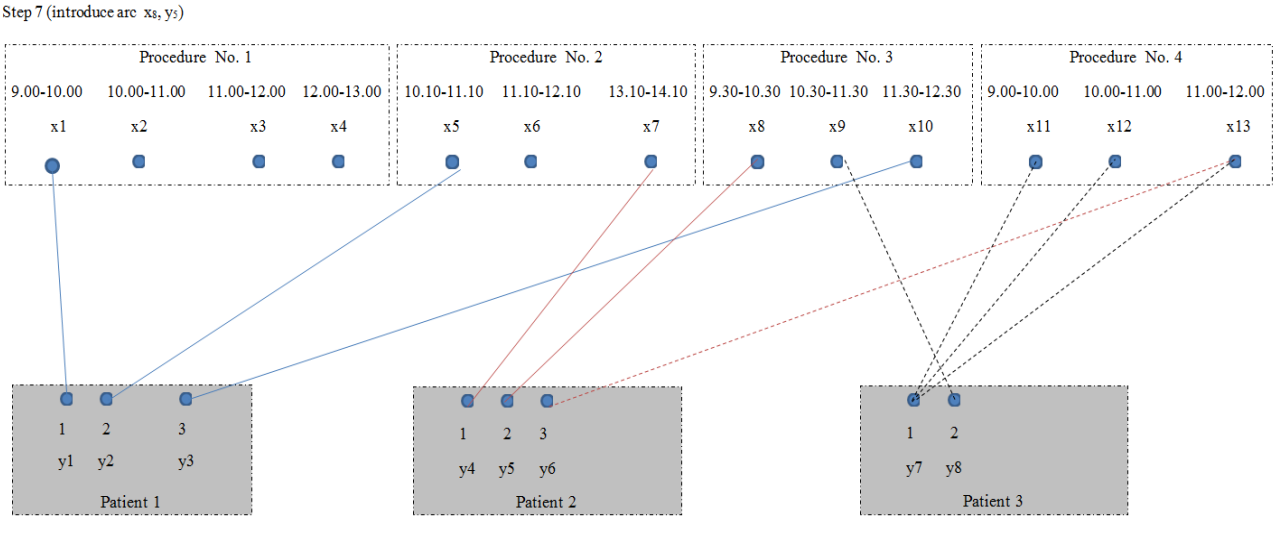

Fig. 10. Introduction of arc (x_8,y_5) to the solution of the problem

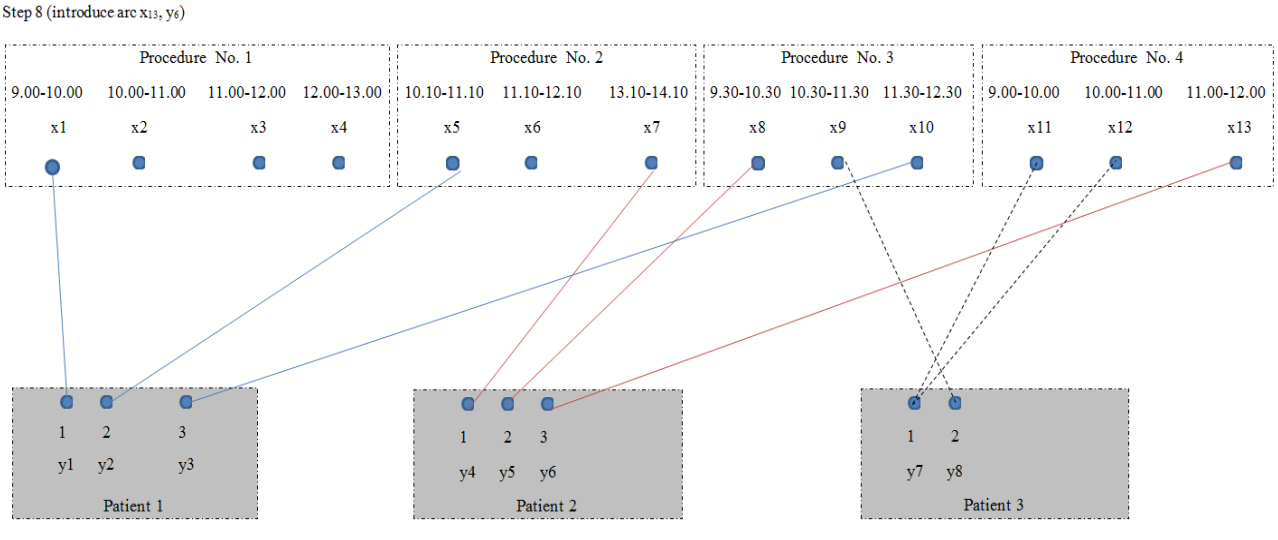

Fig. 11. Selection of arc (x_{13},y_6) and appropriate modification of the bipartite graph

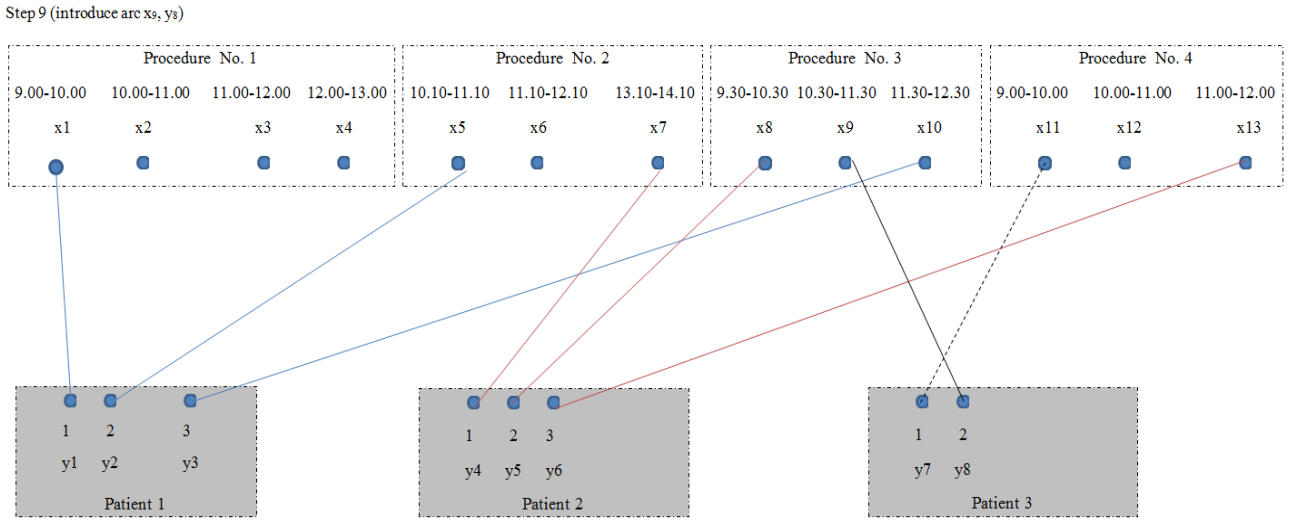

Fig. 12. Selection of arc (x_9,y_8) and appropriate modification of the bipartite graph

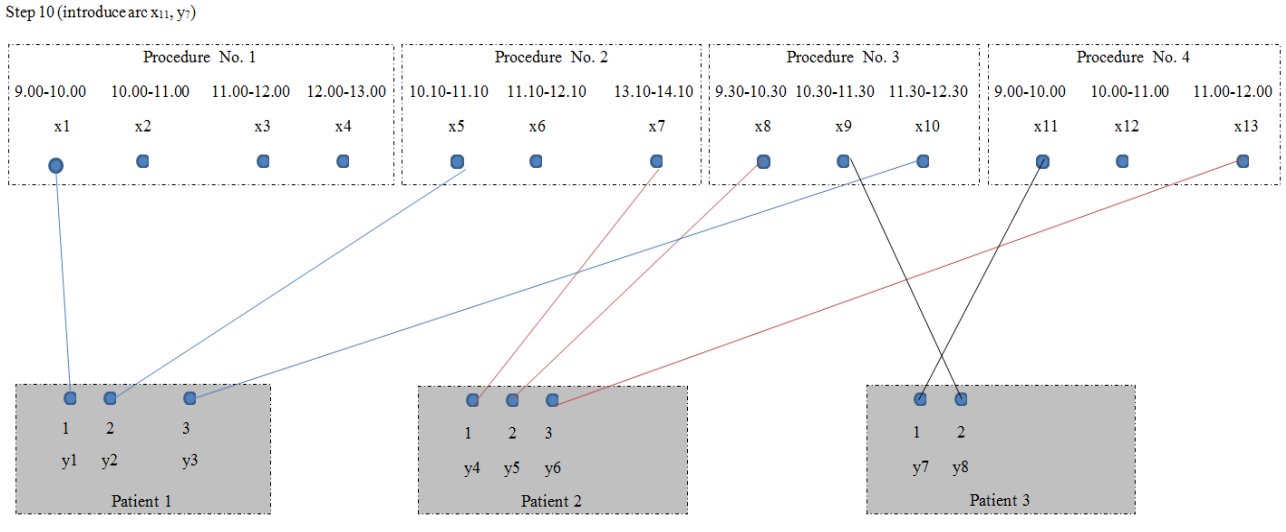

Fig. 13. Selection of arc (x_{11},y_7) and appropriate modification of the bipartite graph

Table 1

Characteristics of computers, which were employed for conducting a comparative computational experiment

No. of configuration	Microprocessor name	Number of cores	Clock frequency	Amount of RAM
1	Pentium 4	1	2.42 GHz	512 MB
2	Intel Celeron E3300	2	2.50 GHz	2 GB
3	Intel Core i5-3570K	4	3.4 GHz	4 GB

The input parameters of experiment (Table 2) were determinate, that is, dependent on the observer, and random, that is, registered, but unguided.

The known constraints in taking the therapeutic procedures (Table 2) are assigned by a set of corollaries C, such as:

$$(x_i, y_j) \rightarrow C_{i,j} = \{(x_{i_1}, y_{j_1}),...,(x_{i_k}, y_{j_k})\}.$$

These constraints take account, first of all, of the impossibility to assign one and the same procedure to different patients for the same time, as well as the compatibility of procedures.

Table 2

List of input parameters of the computational experiment

No. of parameter	Designation	Name of parameter	Type of parameter
1	k_{NP}	Number of therapeutic procedures	Determinate
2	k_p	Number of patients	Random
3	k_{AP}	Number of assigned procedures	Random
4	$C_{i,j}$	Constraints in taking the procedures	Determinate

The initial parameter of experiment is the time t_{BP}, taken for performing the calculations to making up a schedule of taking the therapeutic procedures by patients in the sanatorium with the help of ICS_DENISH.

Accuracy of experimental data depends considerably on the number of carried out tests – the more of them there are, the higher is the authenticity of results. At a small number of factors, it suffices to conduct 50–100 tests at each computational configuration (Table 1).

In each of the 86 conducted tests (Table 3), we measured the time that is needed to perform calculations:

– by the brute force method;

– by the optimum algorithm of solving the problem with disappearing arcs.

Table 3

Results of separate tests in the computational experiment

Number of test	Input parameters			Output parameters (t_{BP}, c)					
				Configuration 1		Configuration 2		Configuration 3	
	k_{NP}	k_P	k_{AP}	t_{BFM1}	t_{OA1}	t_{BFM2}	t_{OA2}	t_{BFM3}	t_{OA3}
1	74	1584	2974	511,24	57,63	365,17	41,16	170,41	19,21
2	74	1312	3217	612,75	96,17	437,68	68,69	204,25	32,06
3	80	1723	3343	712	118,8	508,57	84,86	237,33	39,60
4	80	1784	3578	811,2	141	579,43	100,71	270,40	47,00
5	85	1837	3724	909,2	152,3	649,43	108,79	303,07	50,77
...	...	...	...	...	...	...	...	...	...
86	129	8719	51622	2634,68	587,43	1881,91	419,59	878,23	195,81

Let us build graphs of time dependence on the dimensionality of input parameters (according to the number of test) for the methods, which are compared on three specific configurations (Fig. 14)

The obtained results of computational experiment indicate that

– making up a schedule directly proportionally depends on the number of apexes in a bipartite graph (that is, on the total number of procedures and patients), number of the procedures assigned and constraints

– for the collected initial data, the time for making up a schedule with the use of the brute force method is 8.87 to 4.48 times longer than the time to solve the same problem by the optimum algorithm.

6. Discussion of results of examining the problem on making up a schedule for taking the procedures by patients in a sanatorium

In the process of examining the problem on making up a schedule for taking the procedures by patients in a sanatorium, we stated a mathematical model of the problem on graph matching with disappearing arcs. As demonstrated by subsequent studies, the problem about making up an optimum schedule can be reduced to the extended task of the search for maximum graph matching in a bipartite graph. But the known algorithms, in the classical case, employ for finding the optimal solution the brute force method. In the course of present study, we described an optimum algorithm, which, in contrast to those already known, makes it possible to take into account assigned limitations and it has, in comparison with the brute force method, lower computational complexity due to the reduction in the number of graph matchings, which are analyzed. We have proven the NP-completeness of the problem on graph matching with disappearing arcs.

A practical value of present study is in the possibility of its application in the development and implementation of systems for calendar scheduling and operational management in the therapeutic process. The study is also applicable in the design of control systems for flexible automated systems for the enterprises with discrete character of production.

Authors would like in future to add weight to each arc and to solve the problem on graph matching with "disappearing" arcs in the modified form.

A promising direction of further studies is the modification of known methods (genetic, formic, etc.) to solve the problem on making up a schedule for taking the procedures by patients in a sanatorium.

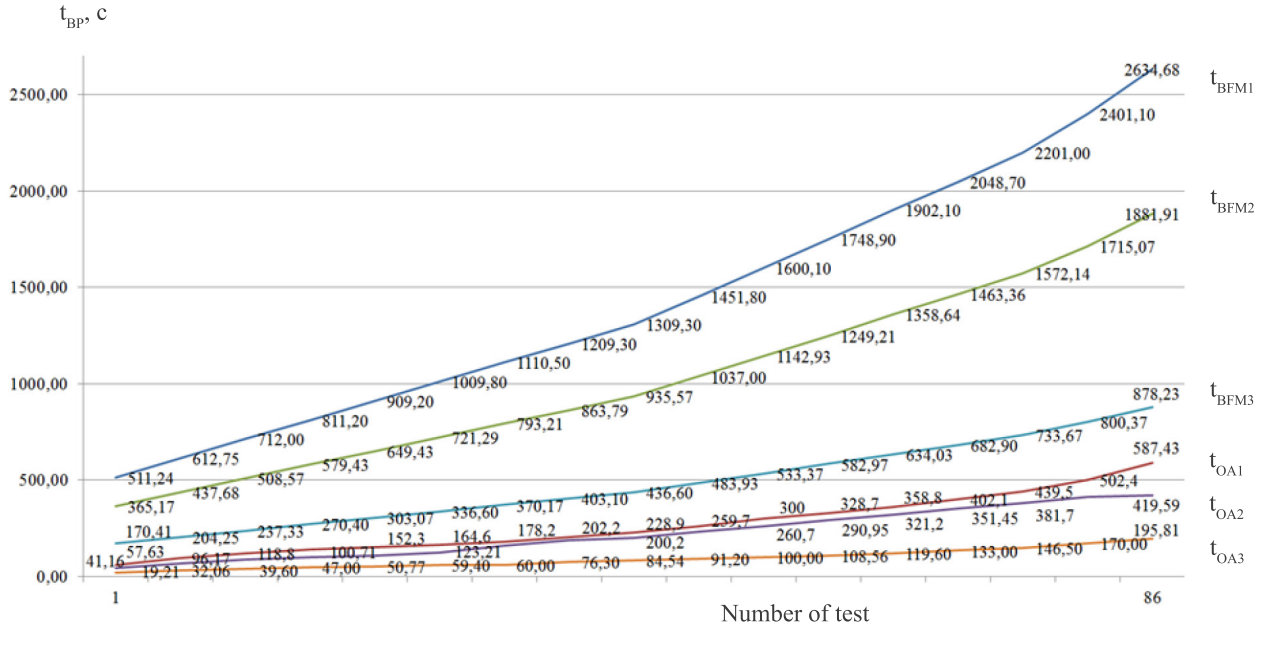

Fig. 14. Graphs of time dependence on the dimensionality of input parameters (according to the number of test) for the methods, which are compared on three specific configurations

7. Conclusions

1. We analyzed the problem on making up a schedule for taking the procedures by patients in a sanatorium. The posed practical problem about making up a schedule for taking therapeutic procedures is stated in the terms of graph theory and, taking into account the assigned limitations, is described by a bipartite graph. It is demonstrated that solving the problem comes down to finding a maximum graph matching in this graph.

2. We performed modification of the problem on graph matching with disappearing arcs for making up a schedule at the assigned constraints. It is proven that the problem "On graph matching with disappearing arcs" is NP-complete in the strong sense.

3. The optimum algorithm for solving the problem about graph matching is developed, whose essence is in finding all maximum graph matchings of the highest power with their subsequent testing for compatibility with the assigned limitations.

4. With the application of the developed software product, we carried out a comparative computational experiment, which made it possible to estimate the time characteristics of optimum algorithm for solving the problem about graph matching with disappearing arcs and to compare them with the time characteristics of the brute force method. According to results of the calculated simulation, the time of making up a schedule under conditions of solving the problem of finding for the largest graph matching on the bipartite graph by the optimum algorithm based on the modified branch and bound method is less than that of the brute force method by 6.8 times.

References

1. Danilchenko, A. Optimal algorithm for solving the problem of matching with vanishing arcs [Text] / A. Danilchenko // Science news. – 2012. – Issue 6. – P. 46–54.
2. Lupin, S. A. The method for solving scheduling problems, focused on cluster computing systems [Text] / S. A. Lupin, T. V. Milehina // Proceedings of the universities. Ser. Electronics. – 2007. – Issue 6. – P. 63–69.
3. Gafarov, E. Hybrid algorithm for solving the problem of minimization of summarized delay for one device [Text] / E. Gafarov // Information technology. – 2007. – Issue 1. – P. 30–37.
4. Tymofijeva, N. K. Solving the problem of scheduling theory planning by structural alphabet search algorithm and hybrid [Text] / N. K. Tymofijeva, V. I. Grycenko // Upravlyayuschye systems and mashiny: Information Technology. – 2011. – Issue 3. – P. 21–36.
5. Panishev, A. The problem of matching with the disappearing "arcs" [Text] / A. Panishev, A. Danilchenko, A. Danilchenko // Modeling and Information Technology. – 2012. – Issue 63. – P. 75–81.
6. Biro, P. Size versus stability in the marriage problem [Text] / P. Biro, D. F. Manlove, S. Mittal // Theoretical Computer Science. – 2010. – Vol. 411, Issue 16-18. – P. 1828–1841. doi: 10.1016/j.tcs.2010.02.003
7. Mugurel, I. Practical Algorithmic Optimizations for Finding Maximal Matchings in Induced Subgraphs of Grids and Minimum Cost Perfect Matchings in Bipartite Graphs [Text] / I. Mugurel // Theory and Applications of Mathematics & Computer Science. – 2014. – Vol. 4, Issue 1. – P. 1–13.
8. Gelain, M. Local Search Approaches in Stable Matching Problems [Text] / M. Gelain, M. Pini, F. Rossi, K. Venable, T. Walsh // Algorithms. – 2013. – Vol. 6, Issue 4. – P. 591–617. doi: 10.3390/a6040591
9. Ageev, A. A. Approximate algorithm for solving the problem of metric peripatetic salesman with an estimate of the accuracy 2 [Text] / A. A. Ageev, A. V. Pjatkin // Discrete Analysis and Operations Research. – 2009. – Vol. 16, Issue 4. – P. 3–20.
10. Sonkin, D. Adaptive algorithm of distributing orders for taxi service [Text] / D. Sonkin // The Tomsk Polytechnic University. – 2009. – Vol. 315, Issue 5. – P. 65–69.
11. Li, W. A DNA Algorithm for the maximal matching problem [Text] / W. Li, E. M. Patrikeev, D. Xiao // Automation and Remote Control. – 2015. – Vol. 76, Issue 10. – P. 1797–1802. doi: 10.1134/s0005117915100070
12. Sergienko, A. Superior appointment scheduler algorithm in heterogeneous distributed computing systems [Text] / A. Sergienko, A. Simonenko, P. Simonenko // System Research and Information Technologies. – 2016. – Issue 2. – P. 20–35.
13. Mazij, O. Recurrent algorithm for solving the problem of the weighted matching [Text] / O. Mazij, A. Morozov, A. Panishev // Cybernetics and Systems Analysis. – 2016. – Vol. 52, Issue 5. – P. 101–112.