Student's Itineraries Through Bachelor Degree Programs in NBU

Nikolay Kirov, Computer Science Department – NBU and
Institute of Mathematics and Informatics – BAS
Petya Assenova, Computer Science Department – NBU

Abstract

The paper presents a mathematical model for a choice of the student's own path of teaching in a Bachelor Program in New Bulgarian University. Every path has to be individual and satisfying the NBU rules. The curriculum is described as a directed acyclic graph with nodes – the courses' and edges – the connections (prerequisites) between courses. The choice of the path by the students is a task for finding itinerary in the graph. The model has a program realization in C++ and input data are for the Bachelor Program *Networking Technologies*. The results and their interpretation are shown too.

1 Background

Every student has to choose for every term m_i courses from M_i offered by a given program, where $m_i < M_i$ and $i = 1, 2, \dots, n$ is the number of the term. The current rules for bachelor programs in NBU are:

- for $i = 1, 2$ (first year), the choice is 3 from 5 ($m_i = 3$, $M_i = 5$);
- for every next term, $i = 3, 4, 5, 6, 7, 8$, the choice is 6 from 8 ($m_i = 6$, $M_i = 8$).

To choose the courses for a term, the student has to read in advance the course passport, where the course syllabus is presented as well as a list of courses as prerequisites.

What are the reasons a student does or does not choose a course proposed for the term?

The reasons not to choose

- The student supposes that he/she already has knowledge and skill which the course offers.
- The student thinks that the course is not useful for him/her.
- The student supposes that the subject of the course is too difficult and he/she could have a problem to take the exam.
- The student considers that the lecturer is not competent in the area and it is pointless to spend the time participating the course.

The reasons to choose

- The student thinks that he/she has an idea about the course but the course will extend his/her knowledge in this area.
- The student thinks that the knowledge and skills obtained in the course will be useful for him/her.
- The student supposes that the subject is easy enough and the corresponding credits could be took without efforts.
- The student considers that the lecturer is very competent in the area and it is worth to learn something from this lecturer.

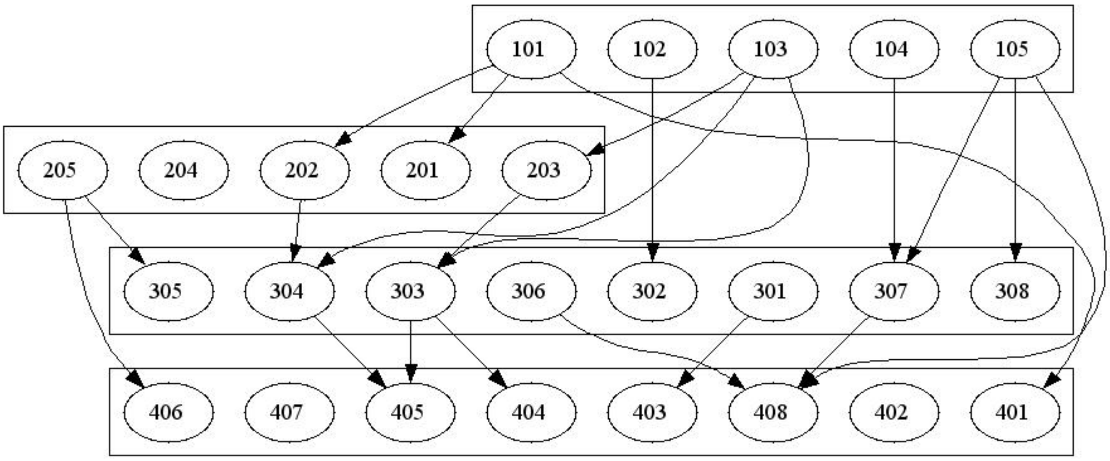

Figure 1: An example of a graph of curriculum for 4 semesters

🔴 Program director does not recommend this course.

🟢 Program director recommends persistently this course.

Some possible criteria which are not very enough working in practice are:

🔴 I cannot enrol this course because I cannot fulfilled the background requirements of the course – I didn't't participate a course from last semester whin is includes in the requirements.

🔴 I do not enrol the course because I can myself learn the material of this course.

🟢 I enrol the course because the background requirements of a course from next semester, which I want to enrol, include this course.

2 Mathematical Model

We suppose in our model that a student can obtain the knowledge and skills the courses from the curriculum only. The lecturer describes for every course the prerequisites which are courses from the previous terms only. The numbers written in nodes are the signatures of the course.

Students itinerary is a list of courses, which the student could enroll satisfying all background requirements.

The curriculum is modeled as a directed acyclic graph (DAG) with nodes the courses' signatures and edges – the connections (prerequisites) between courses (Fig. 1). Every node has a level – term number, when this course is offered. The start node of every edge has less level than the end node.

The curriculum is not a tree because a node may have more than one predicator and the graph may not be connected.

Definition 1. *Itinerary subgraph* is called a subgraph, which contains nodes and all their predicators (Fig. 2). The logic is: If a student wants to enrol a course, he/she has to be already enroled all courses from the minimal itinerary subgraph, which contains this course.

Definition 2. *Nonitinerary subgraph* is called a subgraph, which contains nodes and all their successors (Fig. 3). The logic is: If a student do not enrol a course, then he/she cannot enrol all courses from the minimal nonitinerary subgraph, which contains this course.

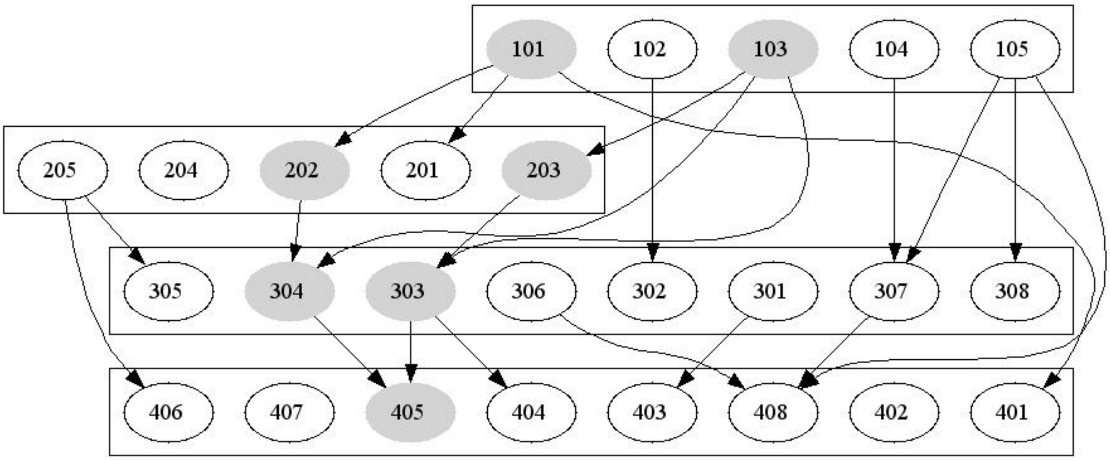

Figure 2: An itinerary subgraph consists of nodes 101, 103, 202, 203, 304, 303 and 405

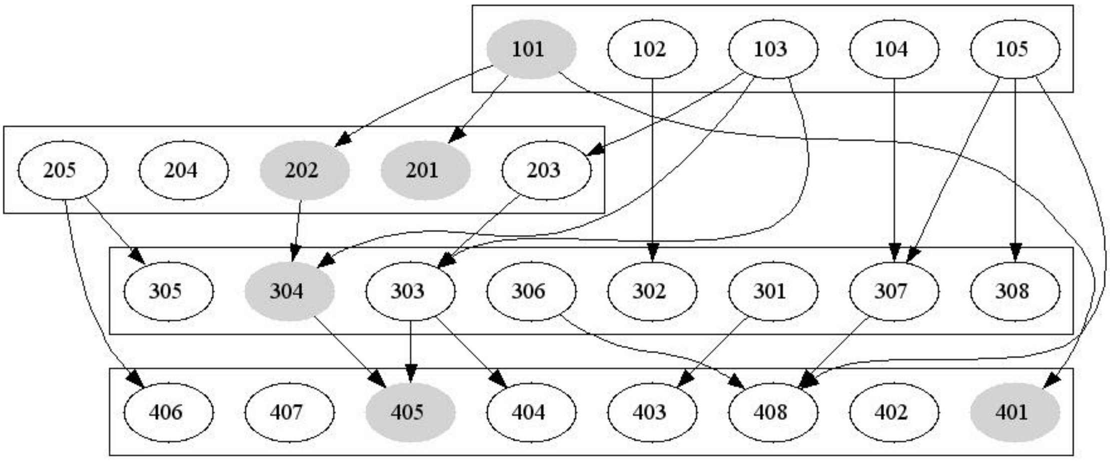

Figure 3: A nonitinerary subgraph consists of nodes 101, 201, 202, 304, 401 and 405

Definition 3. *i-complete itinerary subgraph* is called an itinerary subgraph which contains exact m_i courses of i-th semester. We denote by i-CMIS an i-complete minimal itinerary subgraph (Fig. 4).

Definition 4. *i-complete nonitinerary subgraph* is called a nonitinerary subgraph which contains exact $M - i - m_i$ courses of i-th semester. We denote by i-CMNS an i-complete minimal nonitinerary subgraph (Fig. 5).

Definition 5. *Students itinerary* through a curriculum (or simply *itinerary*) is called an itinerary subgraph, which contains exact m_i nodes of level i, i.e. an i-complete itinerary subgraph for every $i = 1, 2, \ldots, n$ (Fig. 6).

2.1 Tasks

Task 1. For a concrete curriculum find:

a) at least one itinerary through a curriculum, if exists;

b) the number of all itineraries;

c) all itineraries.

Task 2. For a concrete curriculum and a set of chosen courses find:

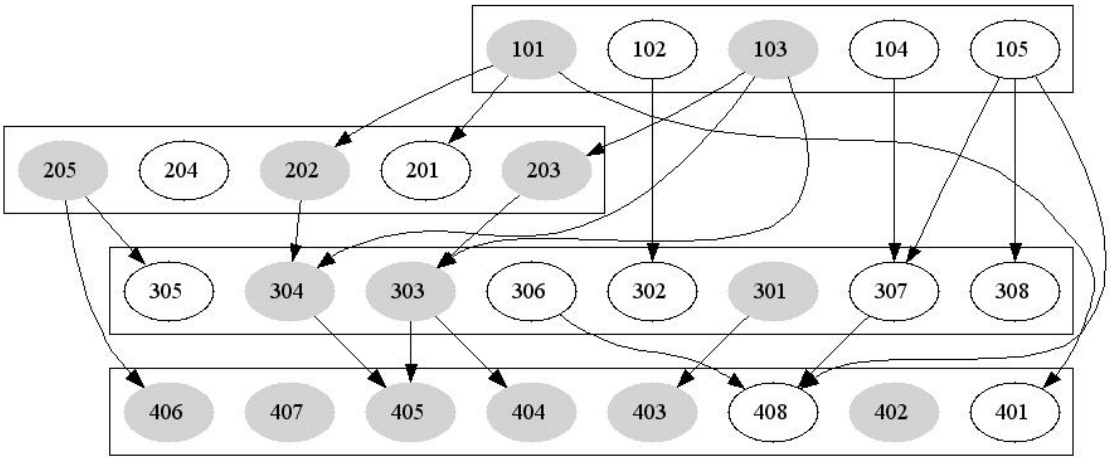

Figure 4: An example of 4-CMIS

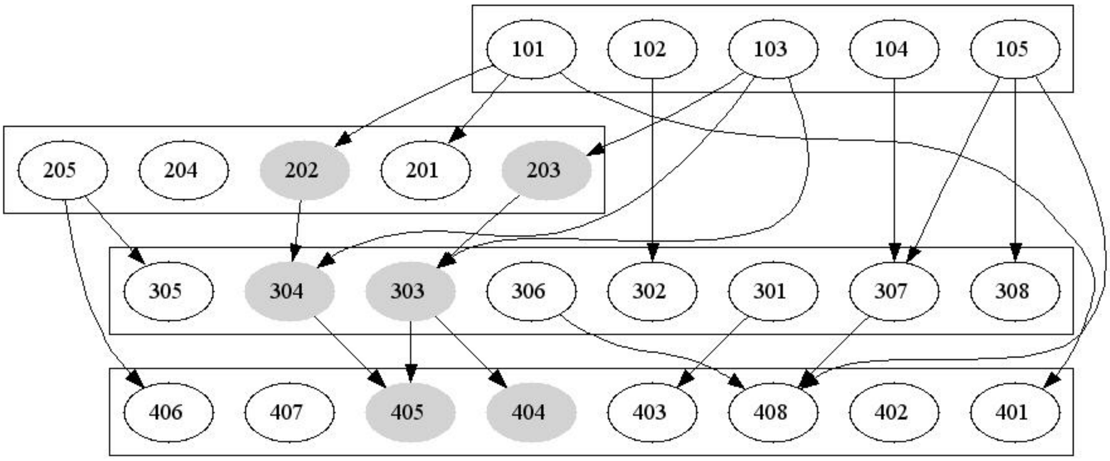

Figure 5: An example of 4-CMNS

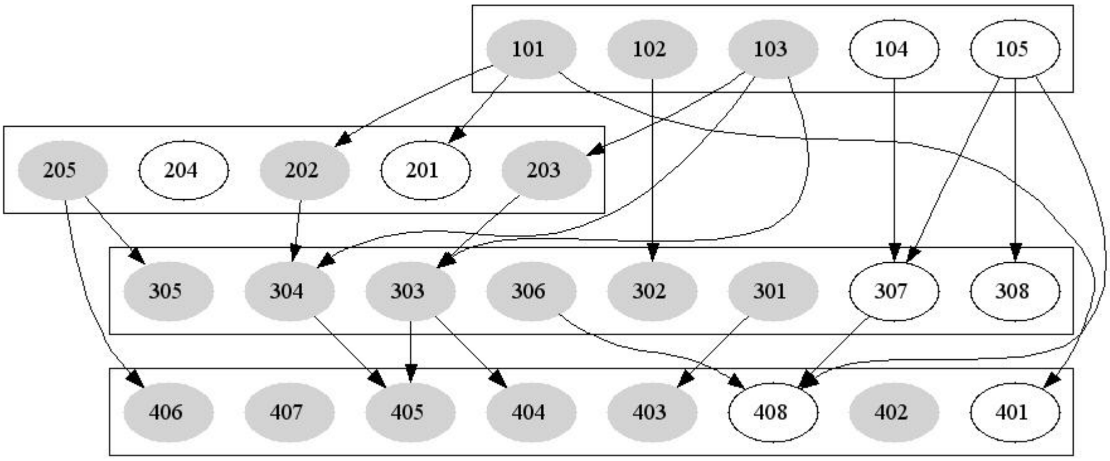

Figure 6: An example of students itinerary

a) at least one itinerary which contains the set, if exists;
b) all itineraries which contain the set.

Task 3. For a concrete curriculum and a set of chosen courses find:
a) minimal itinerary subgraph which contains the set;
b) minimal nonitinerary subgraph which contains the set.
Task 3 can be solved directly.

2.2 Notations

Let a_{ij} be a code of j-th course of term i, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, M_i$. Let $x_{ij} \in \{0, 1\}$ be a variable, which shows whether the course a_{ij} is or is not chosen. For the i-th term a student chooses a subset of m_i elements from the set $\{a_{i1}, a_{i2}, \ldots, a_{iM_i}\}$, i.e. $\sum_{j=1}^{M_i} x_{ij} = m_i$ for every i.

The graph nodes are a_{ij}, when i is the node level. Edges are ordered pairs $(a_{i_1 j_1}, a_{i_2 j_2})$, where $i_1, i_2 \in \{1, 2, \ldots, n\}$, $i_1 < i_2$, $j_1 \in \{1, 2, \ldots, M_{i_1}\}$, $j_2 \in \{1, 2, \ldots, M_{i_2}\}$. To enroll the course $a_{i_2 j_2}$, the student must be have the background knowledge from the course $a_{i_1 j_1}$, i.e. $x_{i_2 j_2} \leq x_{i_1 j_1}$.

Students itinerary is every point from a set, defined by the following equations and inequalities:

- $\sum\limits_{j=1}^{M_i} x_{ij} = m_i$ for every $i = 1, 2, \ldots, n$,
- $x_{i_2 j_2} \leq x_{i_1 j_1}$ for an edge from a_{i_1, j_1} to $a_{i_2 j_2}$
  $1 \leq i_1, i_2 \leq n, \ 1 \leq j_1 \leq M_{i_1}, \ 1 \leq j_2 \leq M_{i_2}$.
- $0 \leq x_{ij} \leq 1$ – integer numbers.

Adding an arbitrary (linear) objective function, we obtain a classic integer programming problem.

2.3 Criteria for Nonexisting Itinerary

(a) Criterion using itinerary subgraph:

Choose m_i courses from a term. Find i-CMIS for this set of courses (Task 3a). If this subgraph has more than m_j nodes for the j-th term ($j < i$), then the set chosen of m_i courses from i-th term cannot be a subset of an itinerary.

() Criterion using nonitinerary subgraph:

For the set of $M_i - m_i$ number of non chosen courses, find i-CMNS (Task 3b). If this subgraph has more than $M_j - m_j$ nodes for j-the term ($j > i$), then the set chosen of m_i courses for i-th term cannot be a set of an itinerary.

The criterion (sufficient condition) is: for a given term, for every choice of the set of m_i courses, it can not be a subset of a student itinerary. If the set A is a i-CMIS for every $i = 1, 2, \ldots, n$, then A is a student itinerary. But the probability of existing such a set is small.

2.4 An Algorithm for Task 1

Let A_{ij} be an i-CMIS. Because of possibility of existing several such subgraphs for a given term, the index j denotes the number of i-CMIS, $(j = 1, 2, \ldots, a_i)$. With $A_{ij}^{(k)}$ we denote the subset of A_{ij} of nodes with level k, $k = 1, 2, \ldots, i$, i.e.

$$A_{ij} = A_{ij}^{(1)} \cup A_{ij}^{(2)} \cup \cdots \cup A_{ij}^{(i)}$$

and $|A_{ij}^{(i)}| = m_i$, but $|A_{ij}^{(k)}| \leq m_k$ for $k = 1, 2, \ldots, i-1$.

Let A be an itinerary. Then A is a union of n sets, which are i-CMIS for $i = 1, 2, \ldots, n$,

$$A = A_{1j_1} \cup A_{2j_2} \cup \cdots \cup A_{nj_n}$$

for some $j_1, j_2, \ldots, j_n$, $j_k \leq J_k$, $k = 1, 2, \ldots, n$. J_i is the number of i-CMIS, which can be subsets of an itineraries.

We have:

$$\begin{array}{rrr}
A_{2j_2}^{(1)} \subset A_{1j_1}, & A_{3j_3}^{(1)} \subset A_{1j_1}, \quad \cdots, & A_{nj_n}^{(1)} \subset A_{1j_1} \\
 & A_{3j_3}^{(2)} \subset A_{2j_2}^{(2)}, \quad \cdots, & A_{nj_n}^{(2)} \subset A_{2j_2}^{(2)} \\
 & \cdots & \\
 & & A_{nj_n}^{(n-1)} \subset A_{n-1j_{n-1}}^{(n-1)}
\end{array} \qquad (1)$$

These are necessary and sufficient conditions for A to be an itinerary.

3 Curriculum of the Bachelor Program *Networking Technologies*

The program *Networking Technologies* is a new bachelor program of Computer Science Department of NBU. Following the rules of NBU, every course must take one term and 30 hours. For that reason some courses are coupled, most often lectures and labs at the same topic. The students are recommended to enroll both courses of the pair. Every course has an identification number (signature) and pairs are connected by &.

The curriculum is divided in two modules: *System Administration* and *Internet Programming*, which are disposed in 3-th and 4-th years. Now the students can choose any course from both modules.

Y	T			C	o	u	r	s	e	s				
1	1	3	5	101&121	102	103&123								
1	2	3	5	151&171	153&173	154								
2	3	6	8	201&221	202&222	206&226	253	294						
2	4	6	8	205&225	254&284	257&277	258&278							
3	5	6	8(14)	274	302	307&317	324&334	341	359	301&311	303	305	321	343
3	6	6	8(11)	352&362	356	357	358&368	371	391	351	353&373			
4	7	6	8(12)	401&411	404&414	409&419	434	435	421	408&418	422			
4	8	6	8(13)	452	454	457&467	474&484	470	464	451&461	458&468	478		

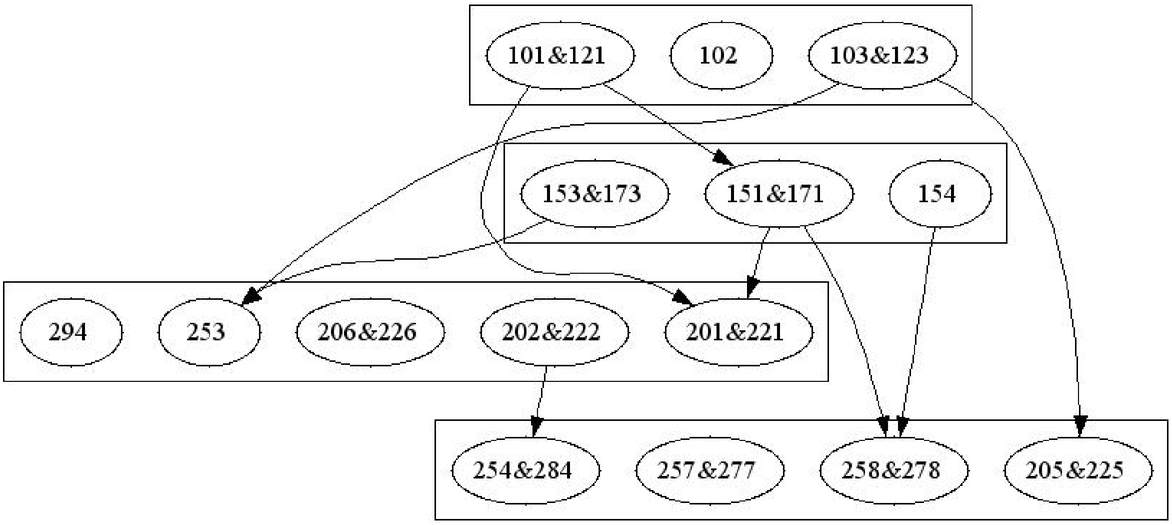

Figure 7: The graph of the Bachelor Program *Networking Technologies* for the first and second years

The courses in the module *System Administration* are in black and red, but those in the module *Internet Programming* are in black and blue. There are 76 courses in the curriculum – 25 double and 26 single, i.e. the graph represented the program has 51 nodes. Edges are 76.

There are two i-CMIS for $i = 1, 2$, when the choice is 3 courses and there are several i-CMIS for $i = 3, 4$, when the choice is 6 courses (Fig. 8).

3.1 Results for curriculum *Networking Technologies*

(a)	(b)	(c)	(d)	(e)	(f)	(g)
1	1	3	5	1	1	1
1	2	3	5	2	2	2
2	3	6	8	2	1	1
2	4	6	8	2	1	1
3	5	6	14	120	22	22
3	6	6	11	20	344	344
4	7	6	12	13	0	876
4	8	6	13	84	0	15 312

(a) Year (b) Term (c) Number of courses to be chosen (d) Number of courses offered (e) Number of i-CMIS (f), (g) Number of itineraries to the corresponding term

There is no itinerary through curriculum *Networking Technologies*! In the seventh term it is impossible to choose 6 courses, as we can see in the column (f) of the table. But if we remove only one edge (from 294 to 401&411), then a lot of itineraries arise – its number is given in the column (g).

4 Requirements to the Lecturers and a Relaxed Task

Every lecturer has to define clearly the courses which are necessary for successful learning the material in its course. If there is no any itinerary, then we can define a relaxed task: Every

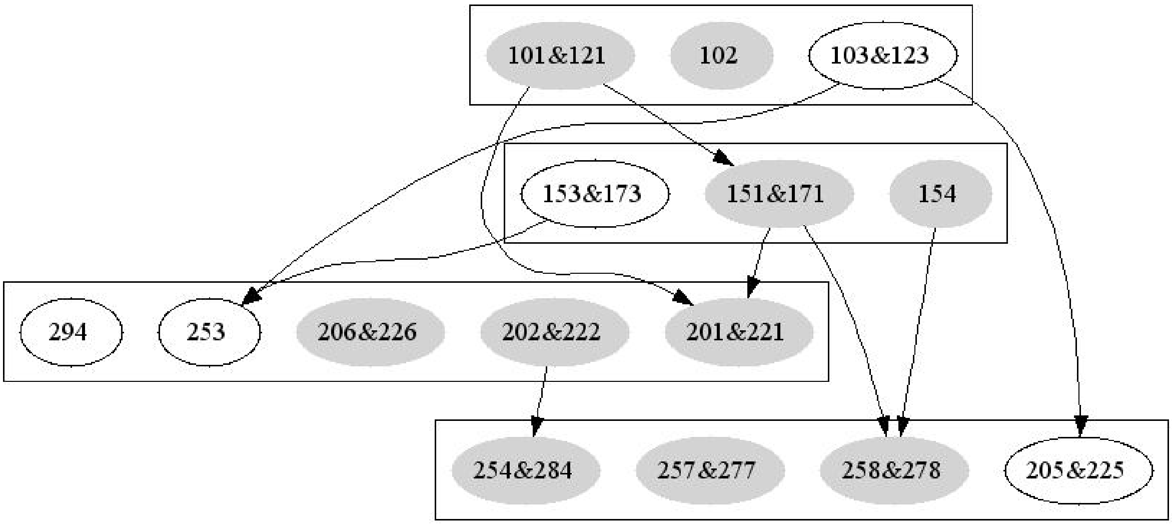

Figure 8: Two i-CMIS for $i = 1, 2$

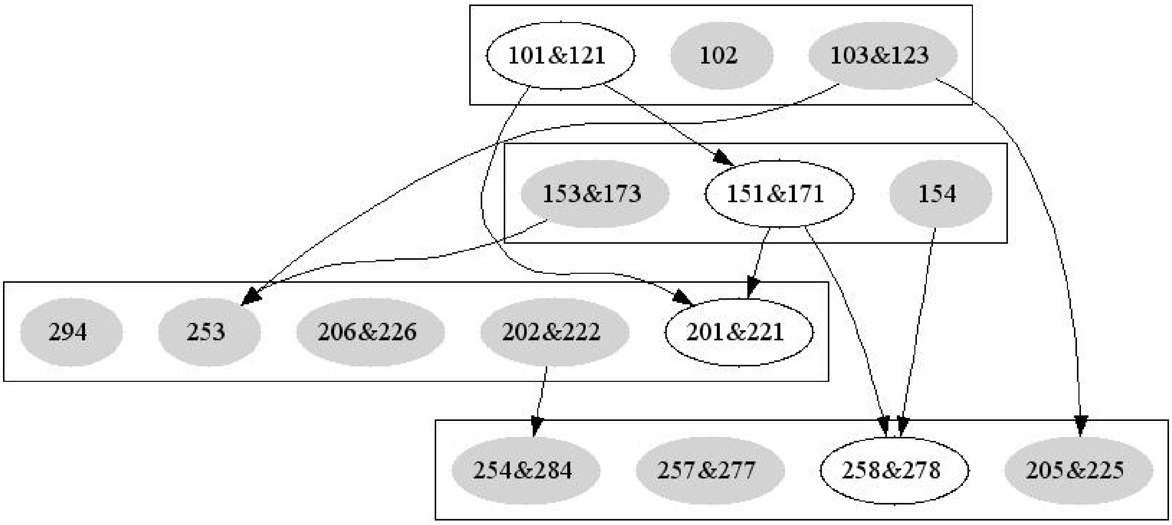

Figure 9: There exist itineraries in those 4 termss (without taking into account the connections with next terms)

lecturer defines two types of preliminary courses: absolutely necessary or recommended. In this case the graph consists of two types of edges necessary and recommended. The tasks can be solved for necessary edges only.

In the choice of several solutions it can be added an optimization criterion using recommended edges.

If there is no solution, the content of some courses should be changed. The decision for these courses could be defined as solutions of optimization problems. For example removing minimal number of edges from the graph in order to obtain at least one solution.

5 The Benefit

The solutions of our model can answer the following questions which are important for curriculum designers:

- How many itineraries exist in a given programme?
- Is a course included in any itinerary?
- How many itineraries includes a given course?
- What is the real choice of a student?
- In case of offering several necessary courses, what is the restriction in the students choice?

For the students the benefits of this model are:

- Every term a student can choose an itinerary or a set of itineraries, which guarantee the possibility of successful completion the program.
- In addition the student can get information about nonitinerary subgraph, which contains the courses which the student does not enroll for a given term. The subgraph consists of all courses which the student cannot enroll all next terms.

References

[1] http://www.nbu.bg/index.php?l=916 (9.06.2007)

[2]