

ИНФОРМАЦИОННЫЙ МЕНЕДЖМЕНТ

Доцент Ротарь Виктор Григорьевич
Фазы IT- администрирования
ПРОЕКТА

Фазы IT- администрирования ПРОЕКТА

1. Формализованное описание, структуризация СМ

- **1.1. Языки описания** (AoN, AoA) - СРС1:
 - генератор проекта (СРС1.1)
 - сеть на языке задач (СРС1.2)
 - сеть на языке событий (СРС1.3)
- **1.2. Логическое упорядочение** СМ:
 - нумерация вершин (ранги, слои) – СРС2,
 - логические ошибки (обрывы, контуры) – СРС3

2. Временной анализ СМ (СРС4)

- Определение продолжительностей задач (СРС4.1)
- Расчет главного параметра проекта – критический путь (СРС4.1)
- Расчет временных параметров событий (СРС4.2)
- Расчет временных параметров задач (СРС4.3)

Фазы IT- администрирования ПРОЕКТА

3. Ресурсное, финансовое планирование (CPC5)

- График Ганта - **CPC5.1**
- Алгоритм Грея - **CPC5.2**
- Частный случай - **CPC5.3**

4. Контроллинг проекта (CPC6)

- Описание проекта программными средствами MS Project (этапы 1-3) – **CPC6.1-CPC6.3**
- Моделирование фазы управления проектом - **CPC6.4**
- Формирование отчетов – **CPC6.5**

1. Фаза структуризации проекта

1.1. Языки описания проекта - СРС1:

- генератор проекта (СРС1.1)
- сеть AoN на языке задач (СРС1.2)
- сеть AoA на языке событий (СРС1.3)

Языки описания проекта

- AoN - Activities on Nodes, сетевая модель с работами в узлах, **сеть на языке работ**
- AoA - Activities on Arrows, сетевая моделью с работами на дугах , **сеть на языке событий**

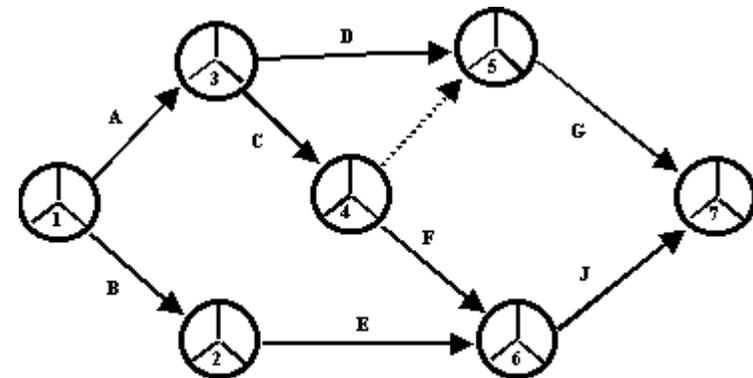
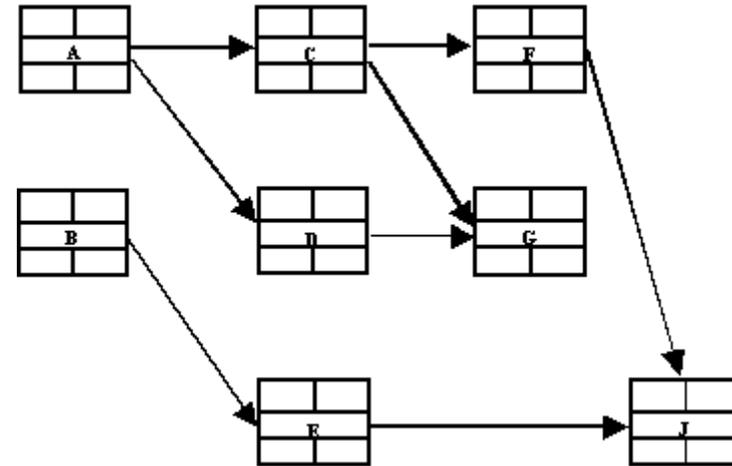


Рисунок 1. Пример сетевого графика модели типа AoA.

ПРИМЕР: СРС1

Формирование индивидуального задания для слушателя INFMAN осуществляется автоматически «генератором проекта» по ФИО слушателя. В матрицу размерностью $(N \times N)$ вписывается его Фамилия Имя и Отчество.

Вписывание ФИО при необходимости производится многократно, начиная с северо-западного угла матрицы, пока не достигнем её юго-восточного угла. При этом последовательно продвигаемся слева -направо и сверху- вниз по наклоненным под углом 45° диагоналям матрицы,.

Здесь N – общее число задач в проекте.



Selection: (1-12)→13→14



СРС1.1: Генератор проекта

Строки 1 - 12: Задачи проекта

Столбцы 1 - 12: Претенденты на предшествование

Столбец 13: Предшествующие задачи

Для задания списка предшествующих задач используется только часть матрицы, находящаяся под главной диагональю. Множество предшествующих задач определяют соответствующие номера столбцов, в клетках которых содержатся гласные буквы в исследуемой части матрицы (см. зеленые клетки).

Столбец 14: Множество непосредственно предшествующих задач.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Р	Т	Ь	Т	И	В	Р	Г	И	К	Е	К	—	—
2	О	Р	К	Р	Е	А	Р	В	И	Ь	И	Е	1	1
3	А	И	Г	Ь	Т	О	Е	В	Р	В	Ь	В	1, 2	2
4	В	Р	Р	О	Т	Ь	Ь	О	Ь	Р	Ь	Г	—	—
5	О	О	Р	К	Р	Р	Г	Р	О	Р	И	О	1, 2	2
6	Г	Ч	И	О	А	И	А	Г	А	Р	Р	К	3, 4, 5	3, 4, 5
7	И	В	Г	Т	Р	Т	И	Т	Г	Ч	И	И	1	1
8	Ь	И	О	Г	О	Р	О	Р	И	В	Р	Е	2, 3, 5, 7	3, 5, 7
9	Р	Р	Р	Р	Г	Р	О	В	Ь	Г	Ь	Р	7	7
10	Ч	О	Ч	Р	Ч	Т	Ь	Р	Р	Р	Ч	А	2	2
11	Т	И	О	И	К	Р	А	О	О	И	Т	Ь	2, 3, 4, 7, 8, 9, 10	4, 8, 9, 10
12	В	Т	В	И	О	Т	Т	Г	В	О	Р	В	4, 5, 10	4, 5, 10

СРС2: Нумерация вершин сети

- Сеть на языке AoA, являющаяся графом Бержа $G=(J,U)$, логически упорядочена (правильно пронумерована), если для каждой дуги $(i, j) \in U$ выполняется условие
$$i < j, \text{ где } i \in J, j \in J (*)$$
- Сеть на языке AoN автоматически является логически упорядоченной «генератором проекта» по условию(*) в соответствии с процедурой формирования задания

ПРОГНОЗ по «INFMAN»



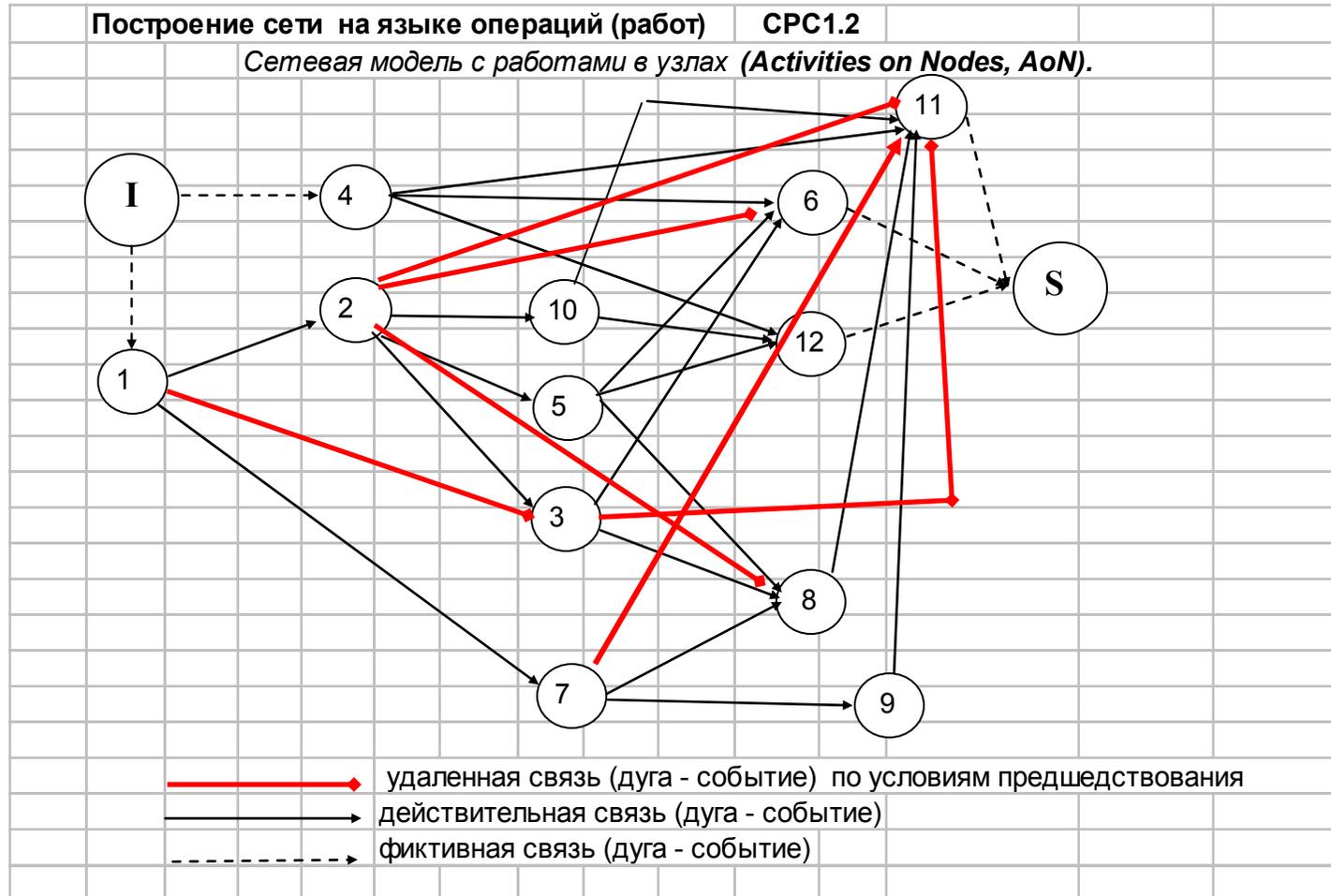
Произошло прерывание при
заполнении матрицы
генератора $N \times N$?

Смотри в Юго-Восточный угол...

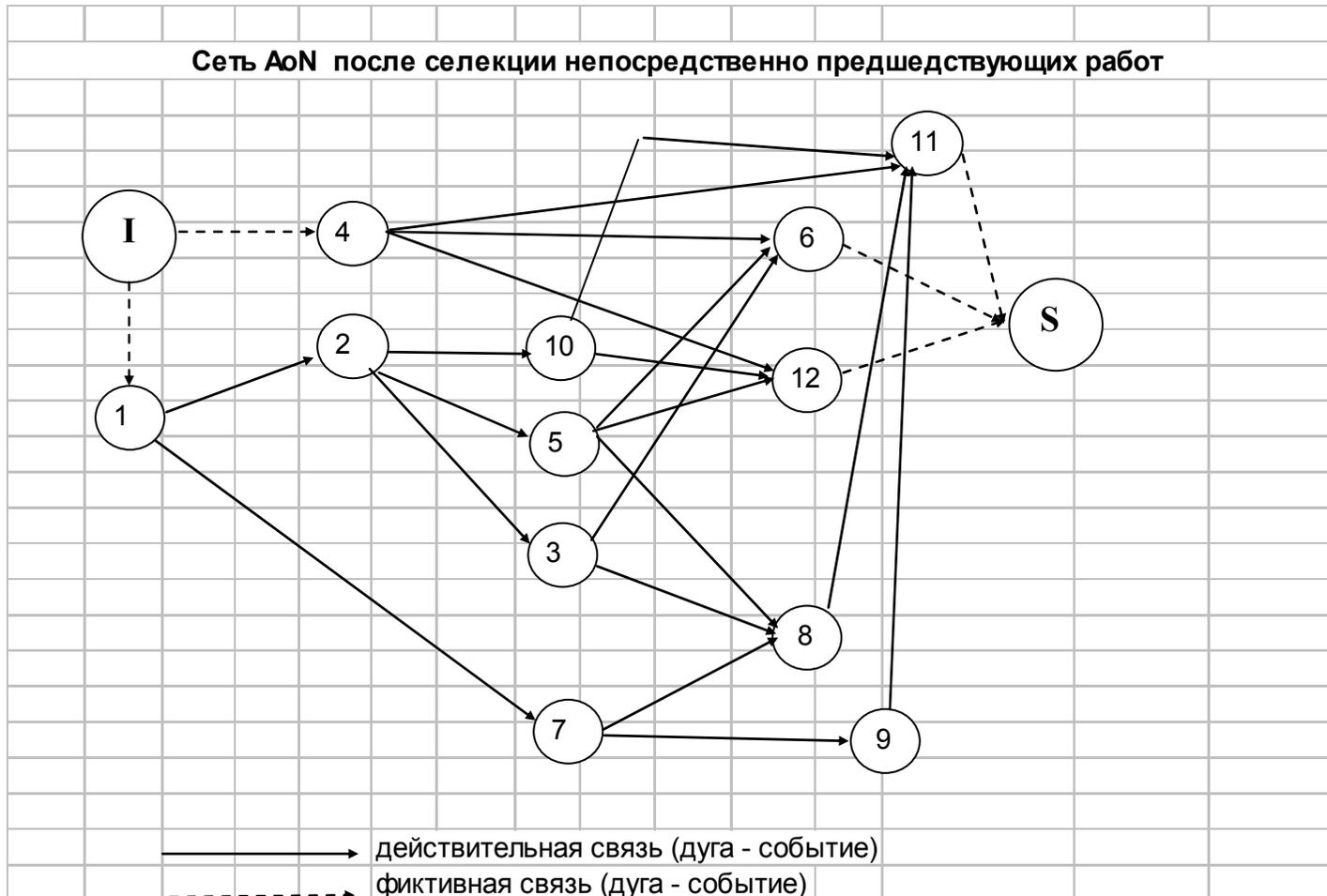
Возможные исходы:

- «**Фамилия Имя и Отчество полностью вошли**» – **Автомат по INFMAN!!!**
- «**Фамилия не вошла**» - изменение соц. положения: замужество, карьерный рост, ...
- «**Имя не вошло**» - расширение состава семьи, сюрпризы от кафедры, деканата, ВУЗа, ...
- «**Отчество не вошло**» - получение наследства, гранта, стипендии, премии, подарка...
- «**Не помните исхода...**» - у Вас вчера был какой-то праздник...

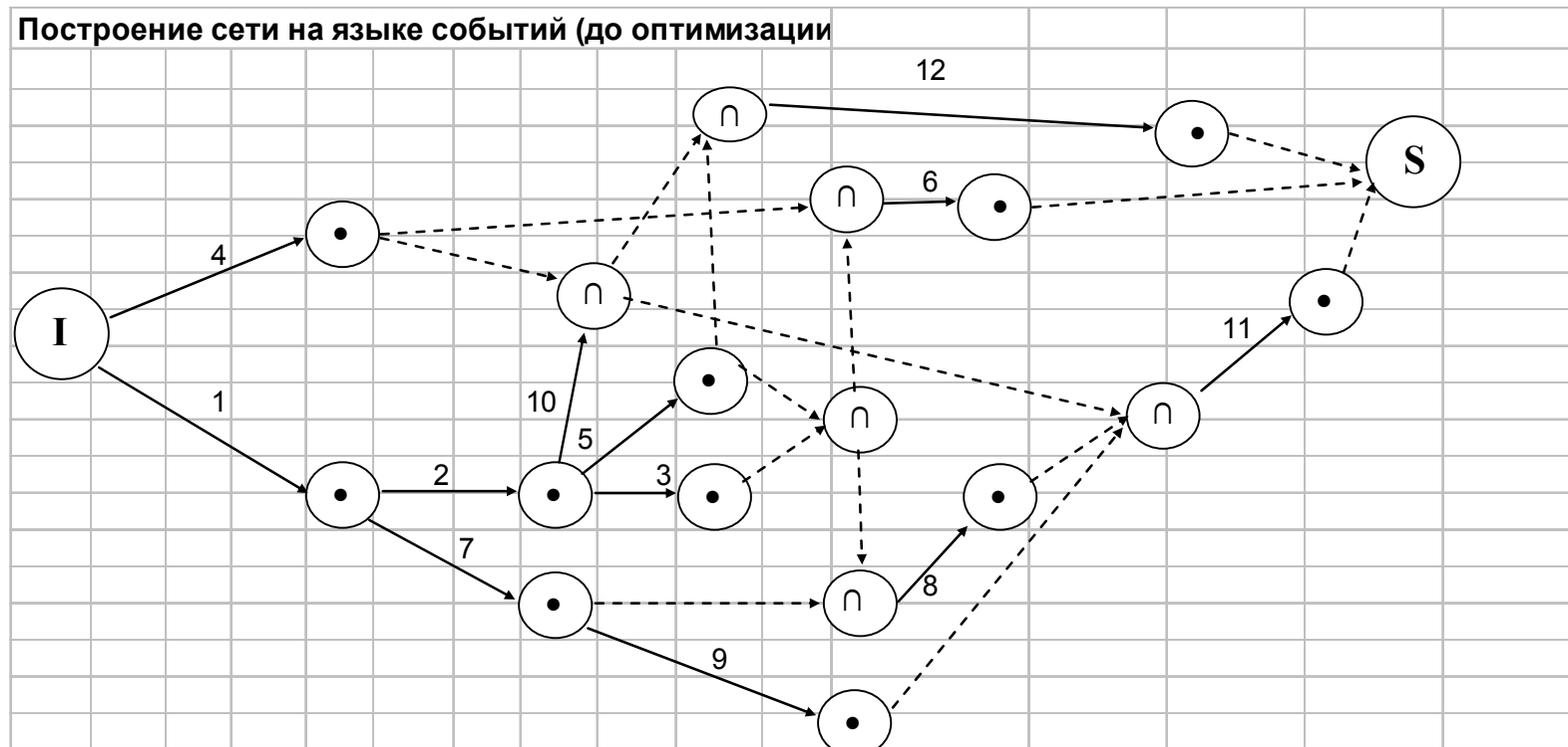
СРС1.2. Сетевая модель с работами в узлах (Activities on Nodes, AoN).



CPC1.2. AoN selection 13→14

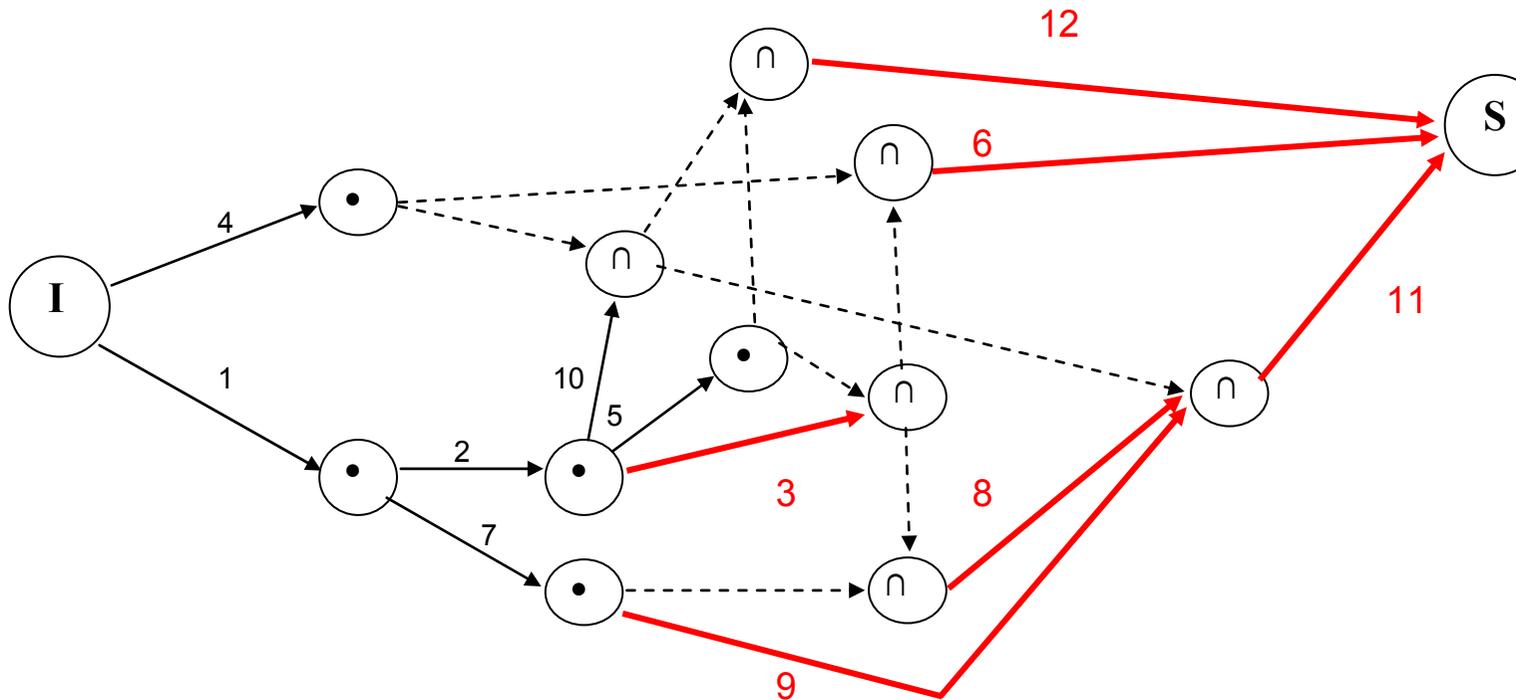


СРС1.3: Сетевая моделью с работами на дугах (Activities on Arrows, AoA)



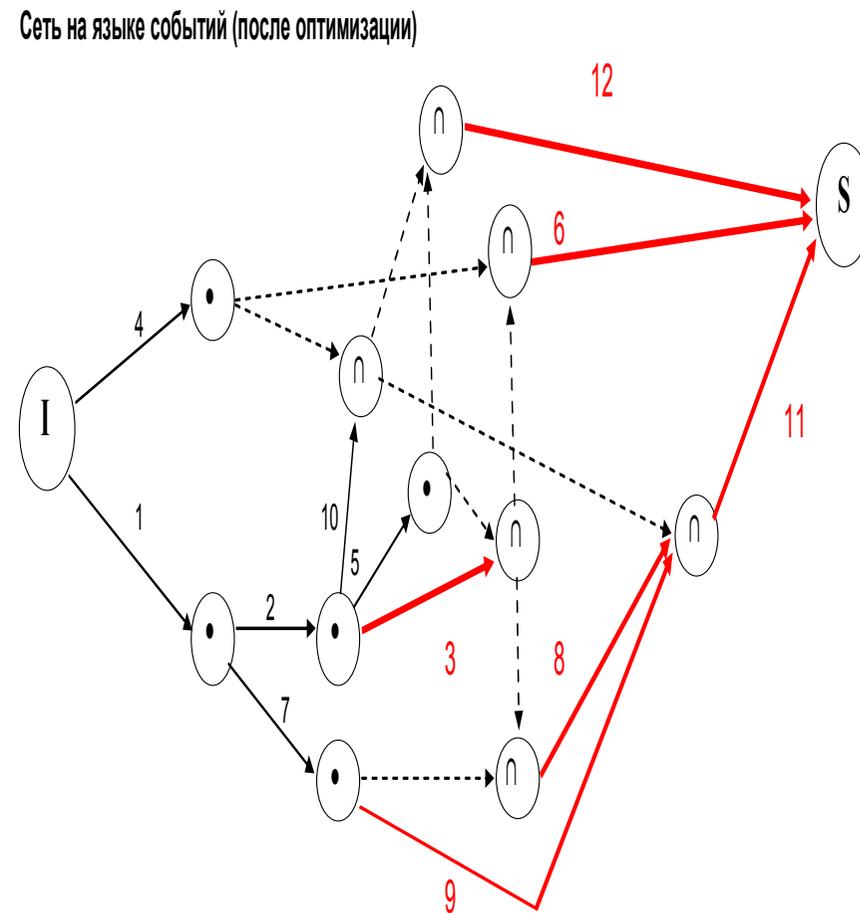
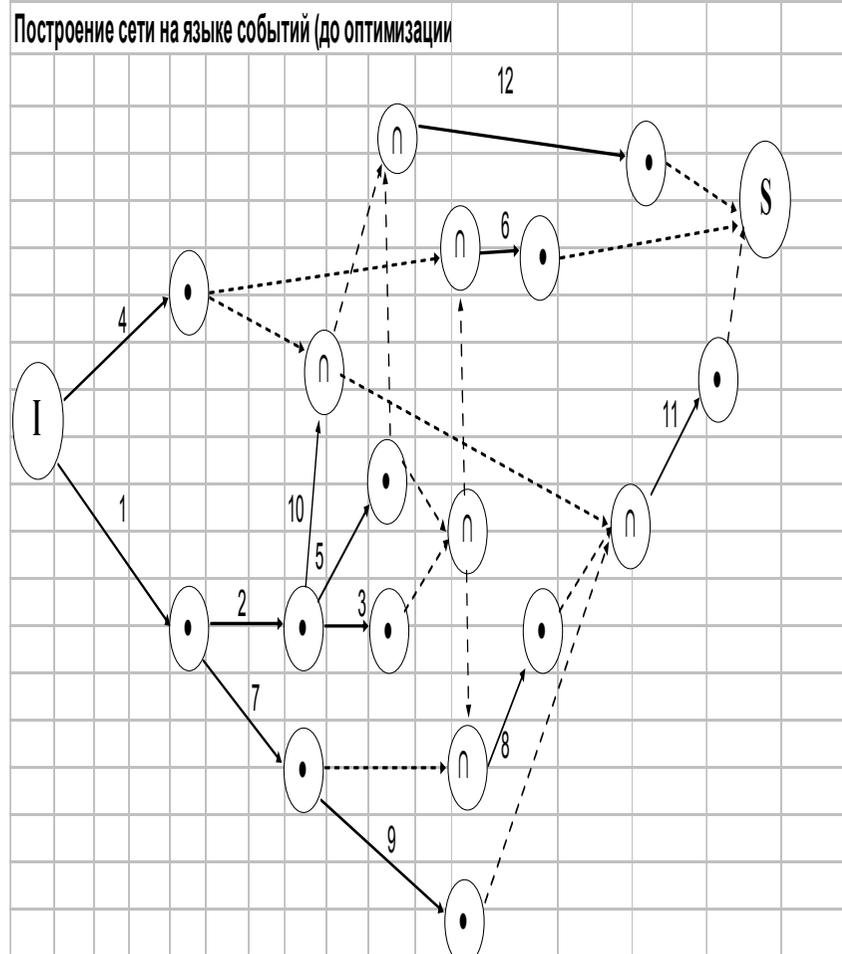
Результаты оптимизации сети AoA

Сеть на языке событий (после оптимизации)

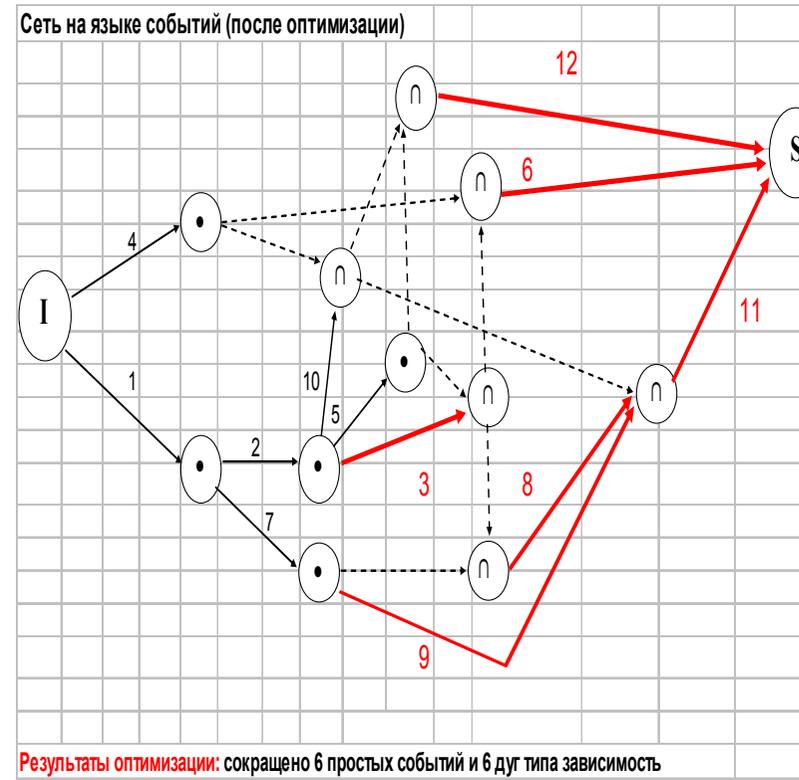
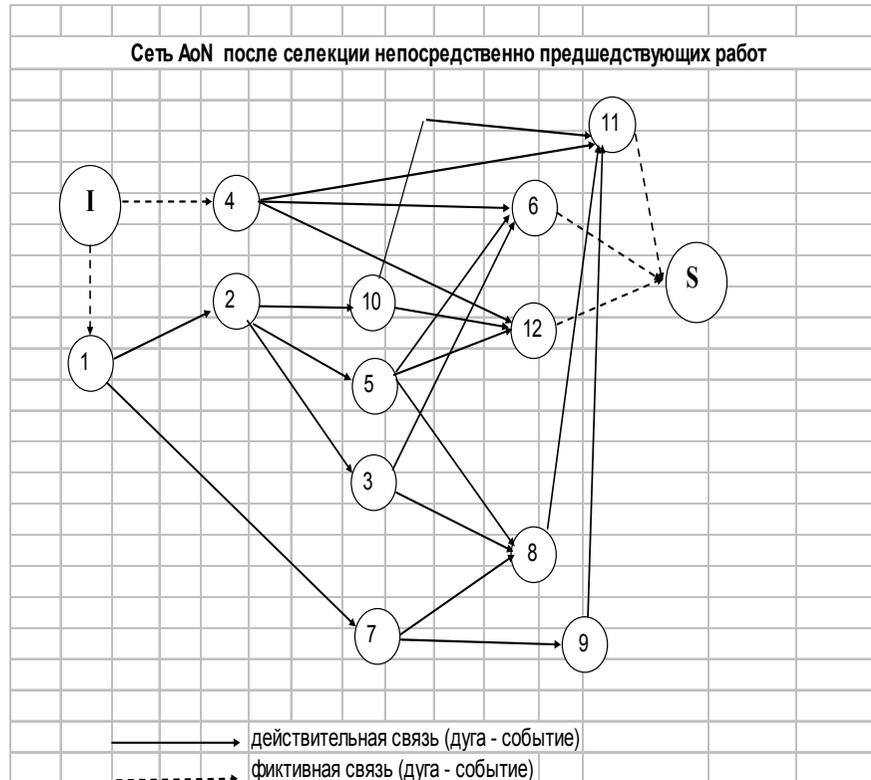


Результаты оптимизации: сокращено 6 простых событий и 6 дуг типа зависимость

Итоги оптимизации сети AoA



Есть предпочтения: AoN \leftrightarrow AoA?



CPC1: That's all !



1.2. Логическое упорядочение сетевой модели (Фаза 1)

- Нумерация вершин (ранги, слои) – **СРС2**,
- Логические ошибки (обрывы, контуры) – **СРС3**

Основные понятия:

- *Логически упорядоченная сеть*
- *Ранг*
- *Слой*
- *Обрыв*
- *Контур*

Метод вычеркивания дуг (МВД)

- **Ранг вершины** — длина максимального по числу переходов (дуг) пути, связывающего данную вершину с истоком сетевой модели проекта
- **Слой графа** — множество вершин, имеющих одинаковый ранг

Алгоритмические схемы МВД

Два варианта разбиения графа $G=(J,U)$ на слои:

а) от истока к стоку $I \rightarrow S$

б) от стока к истоку $S \rightarrow I$

- **Исток сети** - вершина i , для которой выполняется условие $\Gamma_i^{-1} = \emptyset$, где $i \in J$, Γ_i^{-1} - обратное отображение для вершины i

Действия МВД

• **Сток сети** - вершина i , для которой выполняется условие

$$\Gamma_i = \emptyset, \text{ где } i \in J,$$

Γ_i – прямое отображение для вершины i

Вариант А (I → S)

Пусть выполнена $q-1$ итерация. Рассмотрим **q-итерацию МВД:**

Действие q_1 **Формируется** множество вершин $\langle * \rangle$ без **входящих дуг**
(На первой итерации вершина одна - **Исток сети (I)**)

Действие q_2 **Вычисляются** ранги для вершин множества $\langle * \rangle$: $r_*^{(q)} = r_{(q-1)} + 1$

Действие q_3 **Объединяются** в очередной слой вершины множества $\langle * \rangle$, имеющие одинаковый ранг

Действие q_4 **Просматриваются** в произвольном порядке вершины вновь сформированного слоя, **ищутся** исходящие дуги.

Если такие дуги найдены, то все они **вычеркиваются** и **осуществляется переход** на очередную итерацию **q+1** метода, в противном случае – **завершается работа МВД.**

МВД: УСЛОВИЕ ЗАВЕРШЕНИЯ

ОКОНЧАНИЕ для ИТЕРАЦИИ:

У очередной вершины анализируемого слоя при выполнении действия Q_4 не найдено ни одной исходящей дуги.

ВОЗМОЖНЫЕ ИСХОДЫ:

А) **Благополучный** - попали в «**Сток**» сети

В) **Аварийный** - обнаружена логическая ошибка типа «**Обрыв сети**».

«Обрыв сети»: Вершина i , $i \in J$, графа $G=(J,U)$ является «обрывом» сети или «висячей вершиной», если $\Gamma_i = \emptyset$, а вершина не является стоком сети

Алгоритмическая схема МВД ($S \rightarrow I$)

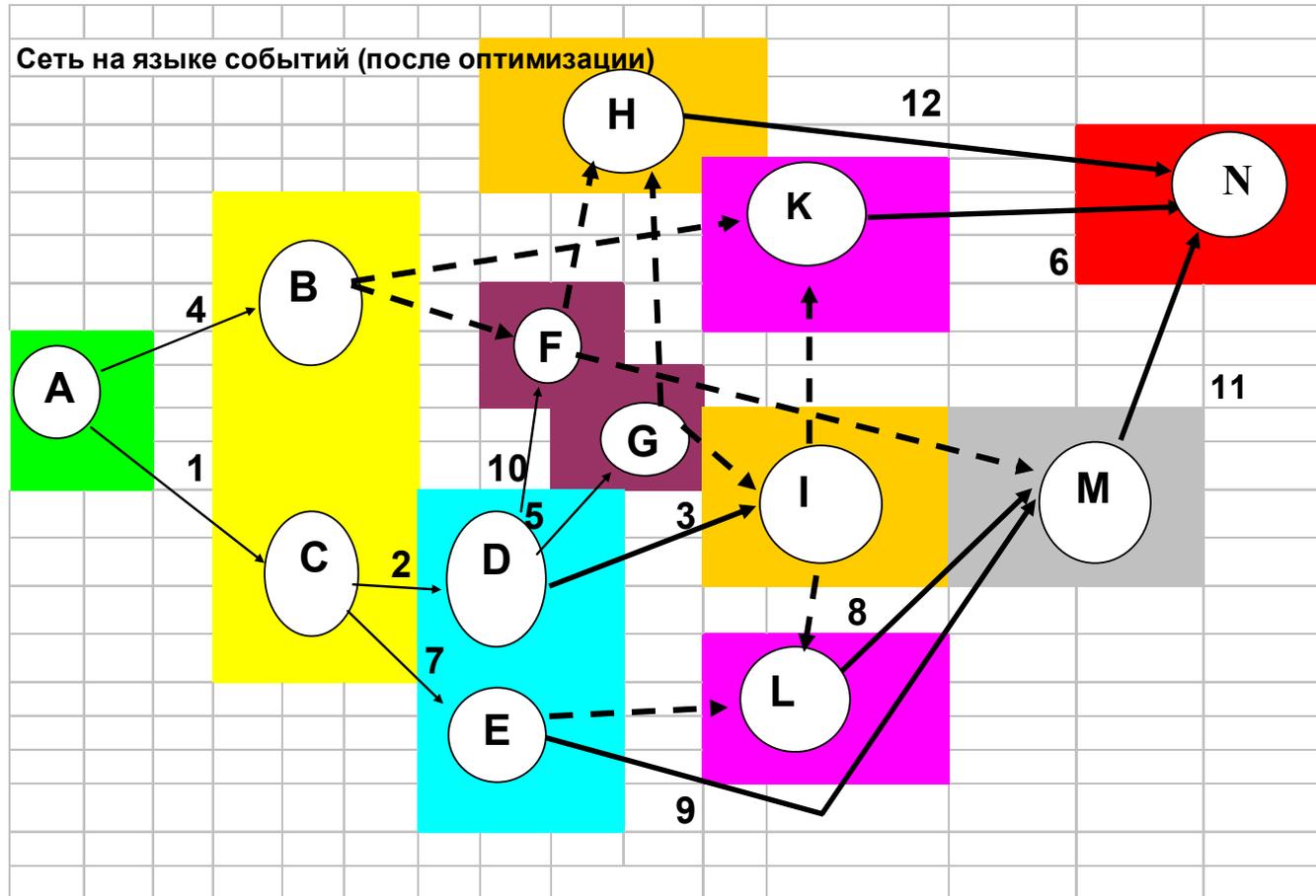
Отличия итерации МВД ($S \rightarrow I$):

- Находятся и обрабатываются вершины **без исходящих** дуг ($q_1 - q_3$). Разбиение на слои осуществляется **в противоположном направлении**: от стока к истоку сети ($S \rightarrow I$)
- Вычеркиваются **входящие** дуги у вершин очередного слоя (q_4)

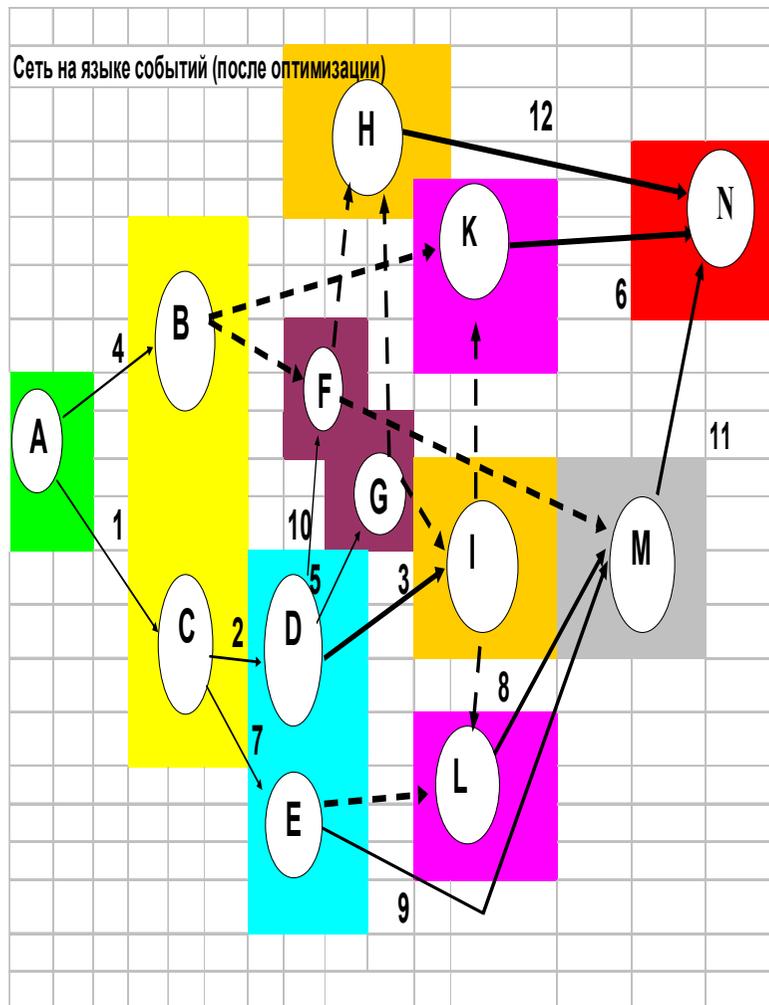
Замечание

Разбиения вершин графа $G=(J,U)$ на слои по схемам $A(I \rightarrow S)$ и $B(S \rightarrow I)$ в общем случае могут не совпадать

ПРИМЕР СРС2: РАНГИ, СЛОИ В МВД



МВД: Действия на матрице смежности вершин



	A	B	C	D	E	F	G	H	I	K	L	M	N
A		1	1										
B										1		1	
C	<i>И</i>			1	1								
D						1	1		1				
E	<i>С</i>										1		1
F								1				1	
G	<i>Т</i>							1	1				
H													1
I	<i>О</i>									1	1		
K													1
L	<i>К</i>											1	
M													1
N					<i>С</i>	<i>Т</i>	<i>О</i>	<i>К</i>					

МВД: Арифметика Демукрона

Начальное условие. Матрица смежности вершин дополняется единицами по главной диагонали.

Нулевая итерация Демукрона

- Вычисляем начальный вектор Демукрона D^0 , где d_i^0 – сумма единиц по i -у столбцу матрицы смежности вершин, $i=1,2, \dots, n$.
- Просматриваем вектор D^0 . К нулевому слою относим те вершины (столбцы), для которых в соответствующих позициях вектора D^0 стоят единицы. На нулевой итерации такой вершиной будет Исток сети (вершина А).

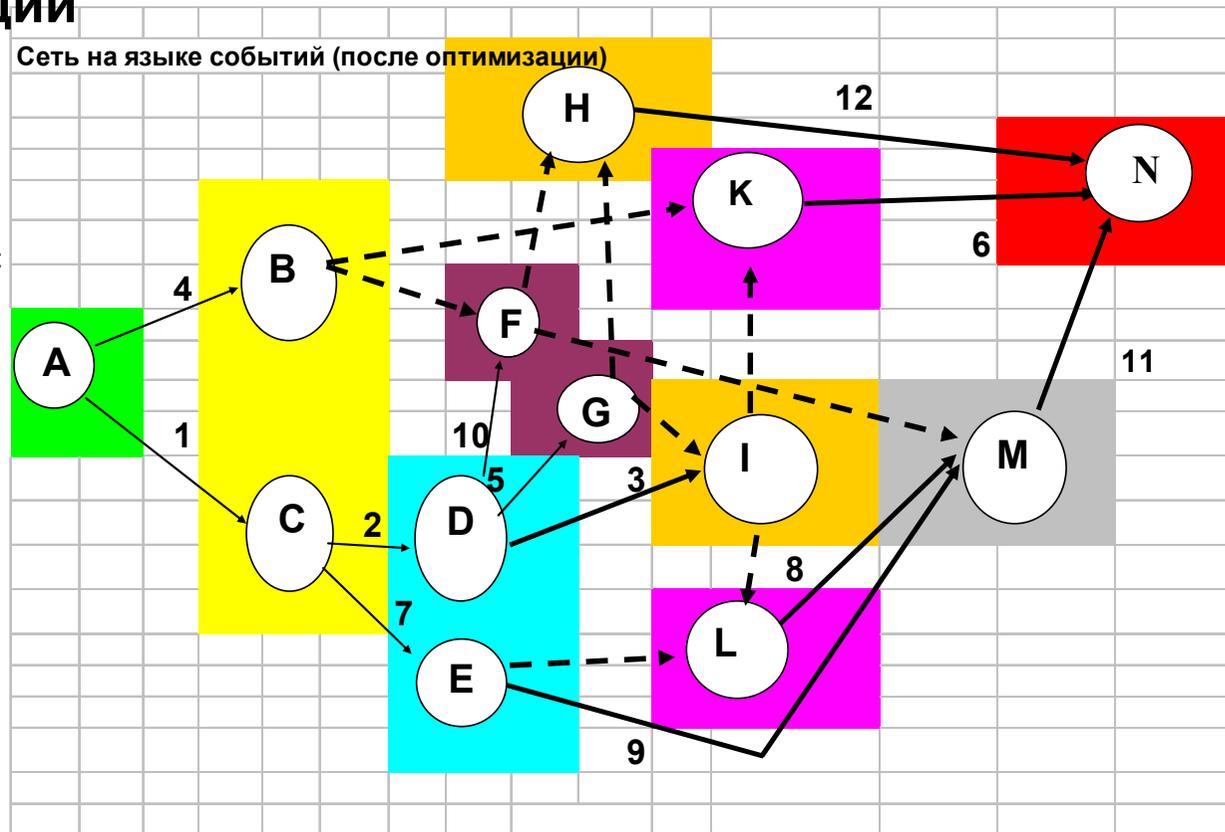
Итерация 1

- Вычисляем вектор Демукрона D^1 :
 $D^1 = D^0 - \Gamma_A$, где Γ_A прямое отображение для вершины А или вектор–строка А матрицы смежности вершин
- 2. Просматриваем вектор D^1 . К первому слою относим те вершины (столбцы), для которых в соответствующих позициях вектора D^1 стоят единицы, таких вершин две: В и С

Арифметика Демукрона

Итерации

- $D^1 = D^0 - \Gamma_A$
- $D^2 = D^1 - \Gamma_B - \Gamma_C$
- $D^3 = D^2 - \Gamma_D - \Gamma_E$
- $D^4 = D^3 - \Gamma_F - \Gamma_G$
- $D^5 = D^4 - \Gamma_H - \Gamma_I$
- $D^6 = D^5 - \Gamma_K - \Gamma_L$
- $D^7 = D^6 - \Gamma_M$



МВД: Алгоритм Демукрона

	A	B	C	D	E	F	G	H	I	K	L	M	N	Р А Н Г И С Л О И	
A	1	1	1												
B		1								1		1			
C	И		1	1	1										
D				1		1	1		1						
E	С				1						1		1		
F						1		1				1			
G	Т						1	1	1						
H								1							1
I	О								1	1	1				
K										1					1
L	К										1	1			
M												1	1		
N					С		Т		О		К		1		
D0	1	2	2	2	2	2	2	3	3	3	3	4	5	A	
D1	0	1	1	2	2	2	2	3	3	3	3	4	5	B,C	
D2	0	0	0	1	1	2	2	3	3	2	3	3	5	D,E	
D3	0	0	0	0	0	1	1	3	2	2	2	3	4	F,G	
D4	0	0	0	0	0	0	0	1	1					H,I	
D5	0	0	0	0	0	0	0	0	0	1	1			K,L	
D6	0	0	0	0	0	0	0	0	0	0	0	1		M	
D7	0	0	0	0	0	0	0	0	0	0	0	0	1	N	

Нумерация вершин

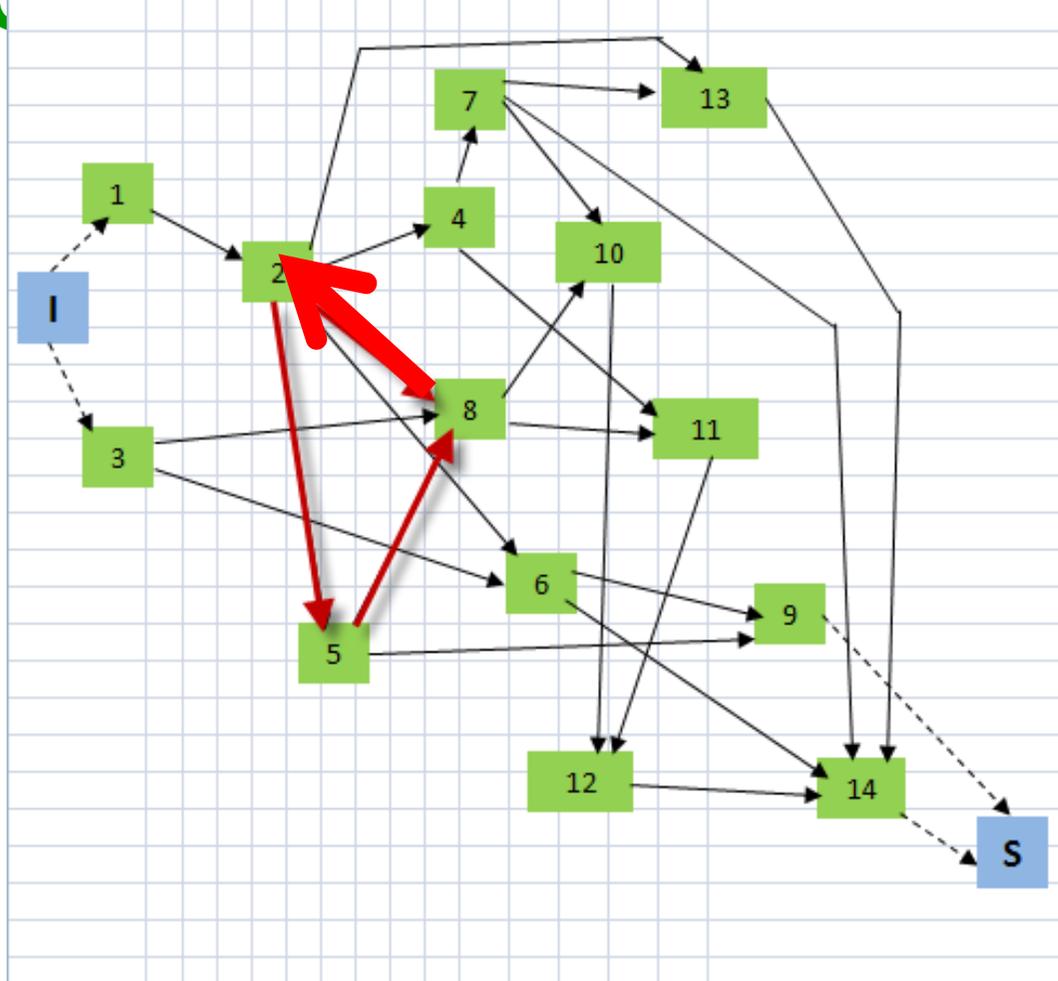


№ п/п	Вершины слоя	Ранг вершины, слой	Номера вершин
1	A	0	100
2	B, C	1	200, 210
3	D, E	2	310, 320
4	F, G	3	420, 430
5	H, I	4	530, 540
6	K, L	5	640, 650
7	M	6	750
8	N	7	850

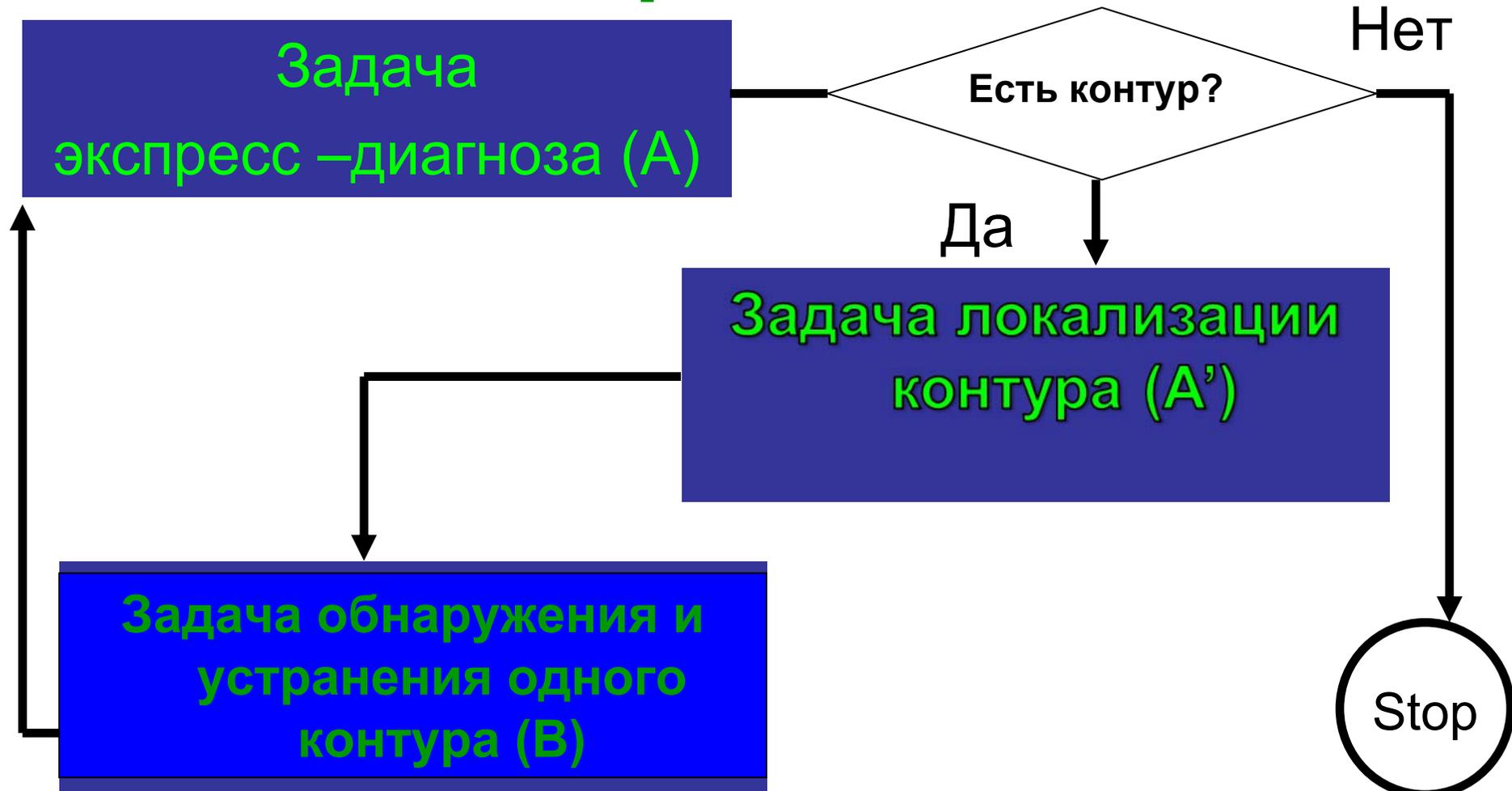
CPC2: That's all !



СРС? - ОШМЕВА ТИПА „УФОНТУР»



Алгоритм АА'В



Исходные положения

Исходная информация о проекте задается в виде списка работ (Табл.1). Результаты вычислений заносятся в таблицу списка вершин проекта (Табл.2).

Для обнаружения контура введем две булевы функции:

$$g(j) = \begin{cases} 1, & \text{если вершина } j \text{ лежит на контуре, либо на пути,} \\ & \text{следующем из контура} \\ 0 & \text{- в противном случае} \end{cases}$$

$$g'(i) = \begin{cases} 1, & \text{если вершина } i \text{ лежит на контуре, либо на пути,} \\ & \text{ведущим в контур} \\ 0 & \text{- в противном случае} \end{cases}$$

ЗАДАЧА «А»

Задача «А»: Экспресс диагноз наличия контуров в проекте.
Диагноз осуществляется по значениям функции $g(j)$, а именно, если её значение равно нулю для каждой вершины j , принадлежащей множеству J графа $G=(J,U)$, то **контуров нет**:

$$g(j)=0, j \in J, G=(J,U)$$

В противном случае мы обнаруживаем контуры по крайней мере хотя бы один. Множество вершин, для которых функция $g(j)=1$, будут образовывать **«прямую тень» контура (движение от $I \rightarrow S$)**:

$$g(j)=1, j \in J', \text{ где } J' \text{ является подмножеством множества } J, \\ G=(J,U)$$

Алгоритмическая схема решения задачи «А». Вычисляется функция $g(j)$ как предел последовательности убывающих рекуррентных функций $g^{(k)}(j)$:

$$g(j) = \lim_{k \rightarrow \infty} \{g^{(k)}(j)\}$$

Итерационный алгоритм решения задачи «А»

0) Нулевая итерация (k=0):

$g^{(0)}(j)=1$, для всех $j \in J$, $G=(J,U)$

Значения функции $g^{(0)}(j)$ заносим в соответствующий столбец итерации табл.2.

Пусть проделана (k-1) итерация и вычислены значения функции $g^{(k-1)}(j)$ для всех $j \in J$, $G=(J,U)$. Рассмотрим действия на общей итерации k.

k) Итерация «k»:

Поочередно (в произвольном порядке, например «сверху вниз») выбираются работы из списка задач U (табл.1).

При выборе очередной работы (i,j) списка в табл.1 производим вычисление значения функции $g^{(k)}(j)$ по следующей рекуррентной формуле:

$$g^{(k)}(j) = \begin{cases} 1, & \text{если } g^{(k-1)}(i)=1, \text{ значение заносится в табл.2 и} \\ & \text{выбираем очередную работу из списка табл.1} \\ \text{не определено} & \text{— в противном случае, в табл.2 ничего не} \\ & \text{заносится и также переходим к выбору очередной работы} \\ & \text{из списка табл.1} \end{cases}$$

Итерационный алгоритм решения задачи «А» (продолжение)

Итерация завершается после просмотра всех работ списка табл.1. Далее **проставляются нули** в тех позициях(вершина-строка) табл.2 (итерация-столбец) , которые соответствуют неопределенным значениям функции $g^{(k)}(j)$.

Условие завершения алгоритма (I->S):

Итерации алгоритма продолжаются до тех пор, пока значения функций $g^{(k)}(j)$ не установятся одновременно для всех вершин графа:

$$g^{(k)}(j) = g^{(k-1)}(j) , \text{ для всех } j \in J, G=(J,U),$$

т.е. пока не будет вычислена функция $g(j)$ (прямая тень контура).

Замечания:

Алгоритм вычисления прямой тени контура эквивалентен алгоритму МВД: нули в столбцах (итерация) табл.2 появляются в строках вершин согласно их рангу. Таким образом, первый ноль появился в строке истока (I, ранг равен 0), следующая серия нулей появилась на второй итерации для двух вершин (1,3 - ранг равен 1), при этом «старые» нули накапливаются и сохраняются. В таком случае процесс разбиения на слои может быть произведен только тогда (пройдем от истока I до стока S), когда в сети нет контуров.

При отсутствии контуров функция $g^{(k)}(j) = g^{(k-1)}(j) = g(j) = 0$ и установится для всех вершин только на итерации k, зависящей от ранга стока проекта, т.е.

$$r(S)+2=k$$

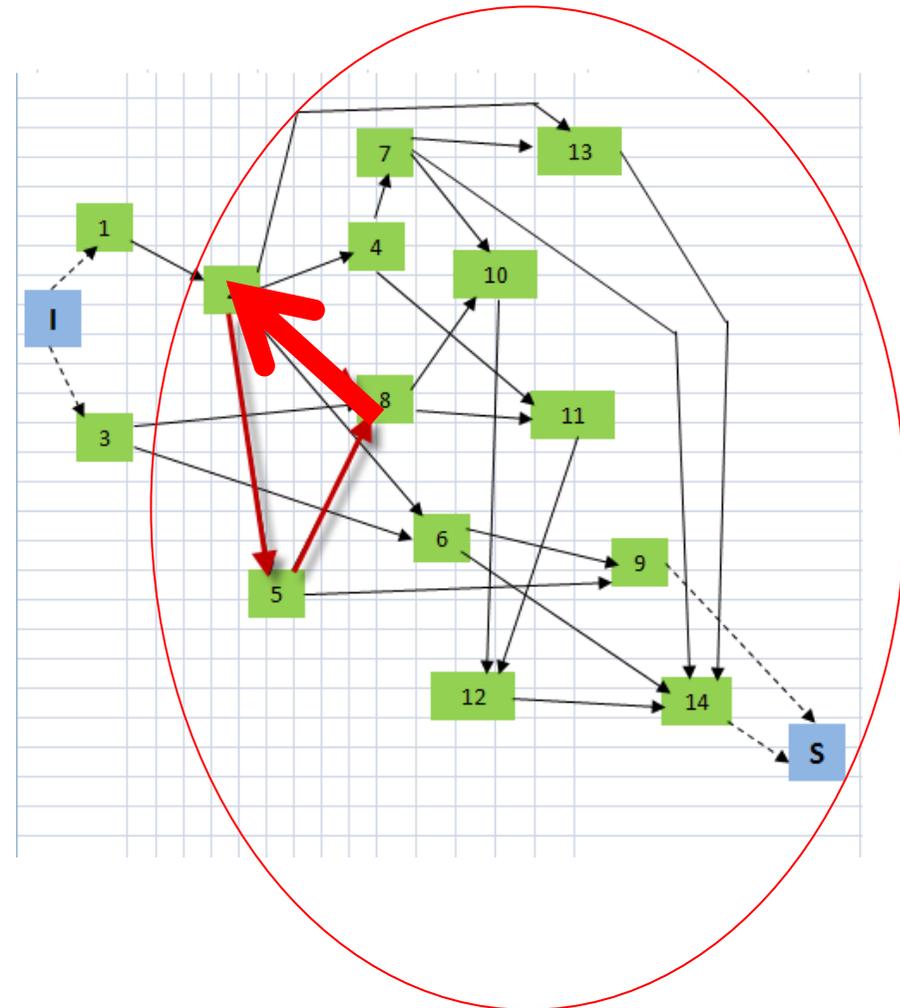
«Прямая тень» контура (задача «А»)

«Прямая тень» контура (эллипс) включает множество вершин, лежащих на контурах и на путях, исходящих (следующих) из контуров и **включает 13 вершин проекта**

В примере «прямая тень» контура определена функцией $g^{(3)}(j)$, вершины, имеющие нулевые значения функции исключаются из тени.

Вывод по задаче А: в сети есть контуры.

Приступаем к их локализации, решаем задачу А'



ЗАДАЧА «А'»- два действия

ЗАДАЧА «А'» - локализации контуров, т.е. определяется множества вершин, лежащих на контурах и путях, соединяющих контуры между собой.

Решение задачи А' включает два действия:

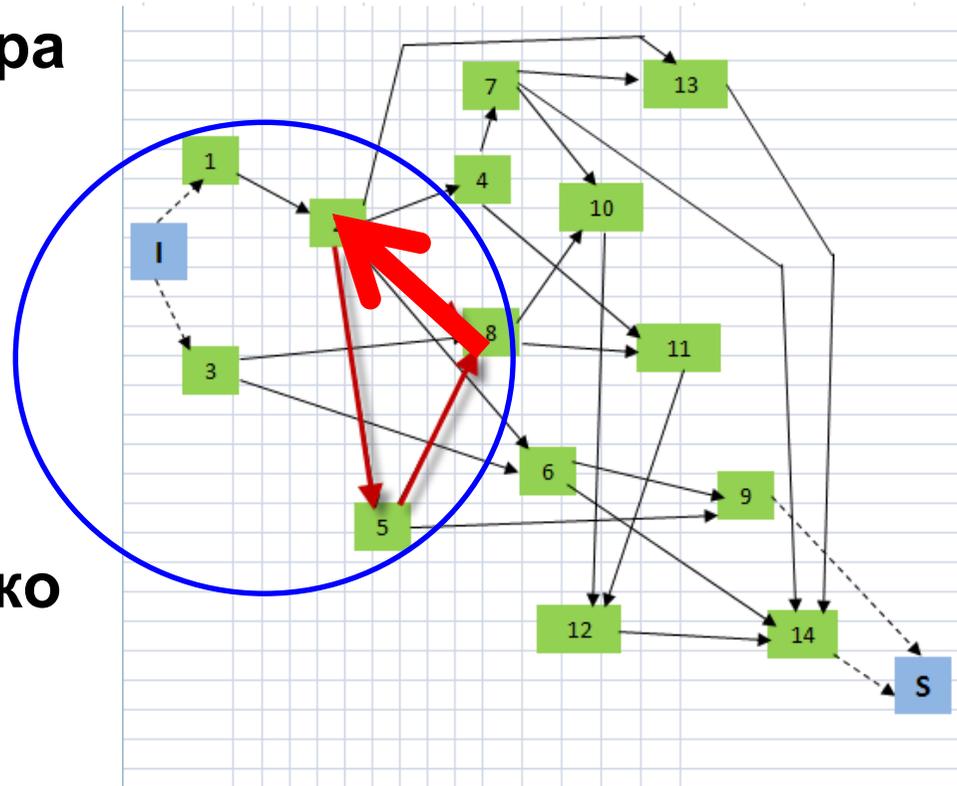
1. **Находим «обратную тень» контуров** – фактически выполняется разбиение графа на слои и вычисляются ранги вершин при движении в направлении $S \rightarrow I$, что соответствует вычислению функции $g'(i)$

$$g'(i) = \lim_{k \rightarrow \infty} \{g^{(k)}(i)\}$$

2. **Локализуем контуры** – находим множество вершин, составляющих пересечение **«прямой»** и **«обратной»** тени контуров

ЗАДАЧА «А'»: «Обратная тень» контура

Алгоритм поиска
«обратной тени» контура
(действие 1, задача А')
аналогичен
рассмотренному ранее
алгоритму решение
задачи А (см. табл.1,
табл.2)
Модифицируется только
само правило
вычисление функции
 $g'(i)$



ДЕЙСТВИЕ 1(A'): модификация правила

$g^{(k)}(i)=$

1, если $g^{(k-1)}(j)=1$, значение заносится в табл.2 и выбираем очередную работу из списка табл.1

не определено – в противном случае, в табл.2 ничего не заносится и также переходим к выбору очередной работы из списка табл.1

Отличие от A: для работы (i,j) по значению функции $g^{(k-1)}(j)$ для вершины j формируем значение функции $g^{(k)}(i)$ для вершины i

В приведенном примере (табл.1, табл.2) разбиение на слои начинается от истока S , значения функции $g^{(k)}(i)$ установились на итерации 6, «новые» нули в столбцах появлялись согласно их рангам вершин при движении в обратном направлении, против ориентации дуг.

«Обратную тень» контура составляют шесть вершин (1, 1, 2, 3, 5, 8)

ДЕЙСТВИЕ 2(A'): Локализация контуров

Находим пересечение **Прямой** и **обратной** теней контура (Действие 2 задачи «A'»).

Формируем множество вершин A^* , локализующих контуры следующим образом:

$$A^* = \{ i / g(i) = g'(i) = 1, i \in J, G = (J, U) \}$$

Для нашего примера множество

$$A^* = \{3; 5; 8\}$$

В случае, когда контур единственный в сети, операция по локализации контура (задача A') фактически дает ответ и по задаче B

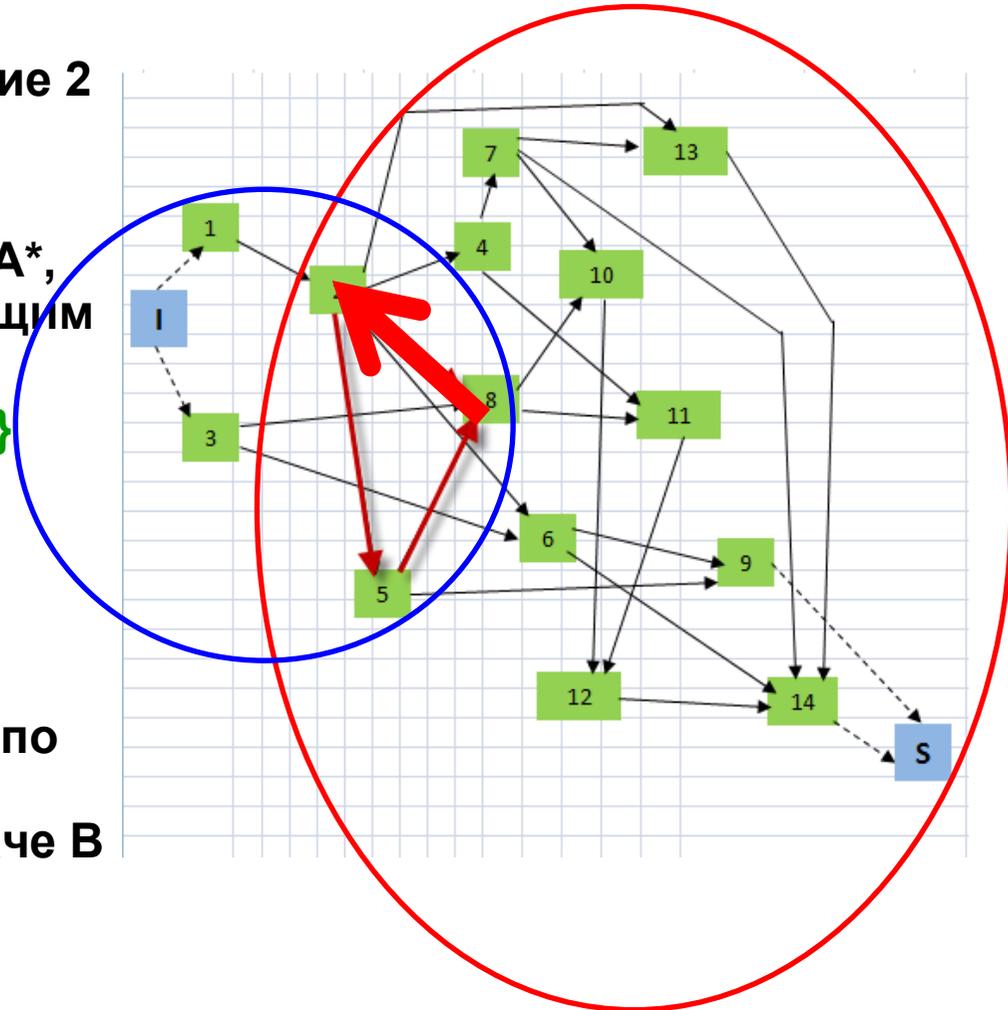


Табл.1

N п/п	(i,j)	№ п/п/ (i,j)
1	(1,1)	15 (8,10)
2	(1,3)	16 (8,11)
3	(1,2)	17 (9,S)
4	(8,2)	18 (10,12)
5	(3,6)	19 (11,12)
6	(3,8)	20 (10,12)
7	(4,7)	21 (12,14)
8	(4,11)	22 (13,14)
9	(5,8)	23 (14,S)
10	(5,9)	
11	(6,9)	
12	(6,14)	
13	(7,10)	
14	(7,13)	

СРСЗ: АЛГОРИТМ «А-А'-В»

Табл.2

I=>S S=>I A*

j	g0	g1	g2	g3	g' 0	g' 1	g' 2	g' 3	g' 4	g' 5	g' 6	A*	(i,j)	
1	1	0	0	0	1	1	1	1	1	1	1		(1,1)	
1	1	1	0	0	1	1	1	1	1	1	1		(1,3)	
2	1	1	1	1	1	1	1	1	1	1	1	+	(1,2)	
3	1	1	0	0	1	1	1	1	1	1	1		(2,4)	
4	1	1	1	1	1	1	1	1	1	1	0		(2,5)	*(1)
5	1	1	1	1	1	1	1	1	1	1	1	+	(2,6)	
6	1	1	1	1	1	1	1	0	0	0	0		(2,13)	
7	1	1	1	1	1	1	1	1	1	0	0		(3,6)	
8	1	1	1	1	1	1	1	1	1	1	1	+	(3,8)	
9	1	1	1	1	1	1	0	0	0	0	0		(4,7)	
10	1	1	1	1	1	1	1	1	0	0	0		(4,11)	
11	1	1	1	1	1	1	1	1	0	0	0		(5,8)	*(2)
12	1	1	1	1	1	1	1	0	0	0	0		(5,9)	
13	1	1	1	1	1	1	1	0	0	0	0		(6,9)	
14	1	1	1	1	1	1	0	0	0	0	0		(6,14)	
S	1	1	1	1	1	0	0	0	0	0	0		(7,10)	
													(7,13)	

ЗАДАЧА «В»

ЦЕЛЬ ЗАДАЧИ «В»: выявить и устранить только один из возможных контуров.

Действия алгоритма «В»:

Действие В1: На основе множества вершин A^* сформировать множество работ U^* , локализирующих контуры

$U^* = \{ (i,j) \in U, G=(J,U) / (i \in A^*) \cap (j \in A^*) \}$ **Для примера: формируем множество $U^* = \{(8,2); (5,8); (2,5)\}$**

Действие В2: Выявить один из возможных контуров.

В списке задач U^* выбирается произвольная работа $(i,j) \in U^*$ и строится цепочка переходов пока не будет обнаружено её замыкание/ **Для примера: строим цепочку $(8,2) \Rightarrow (2,5) \Rightarrow (5,8)$;**

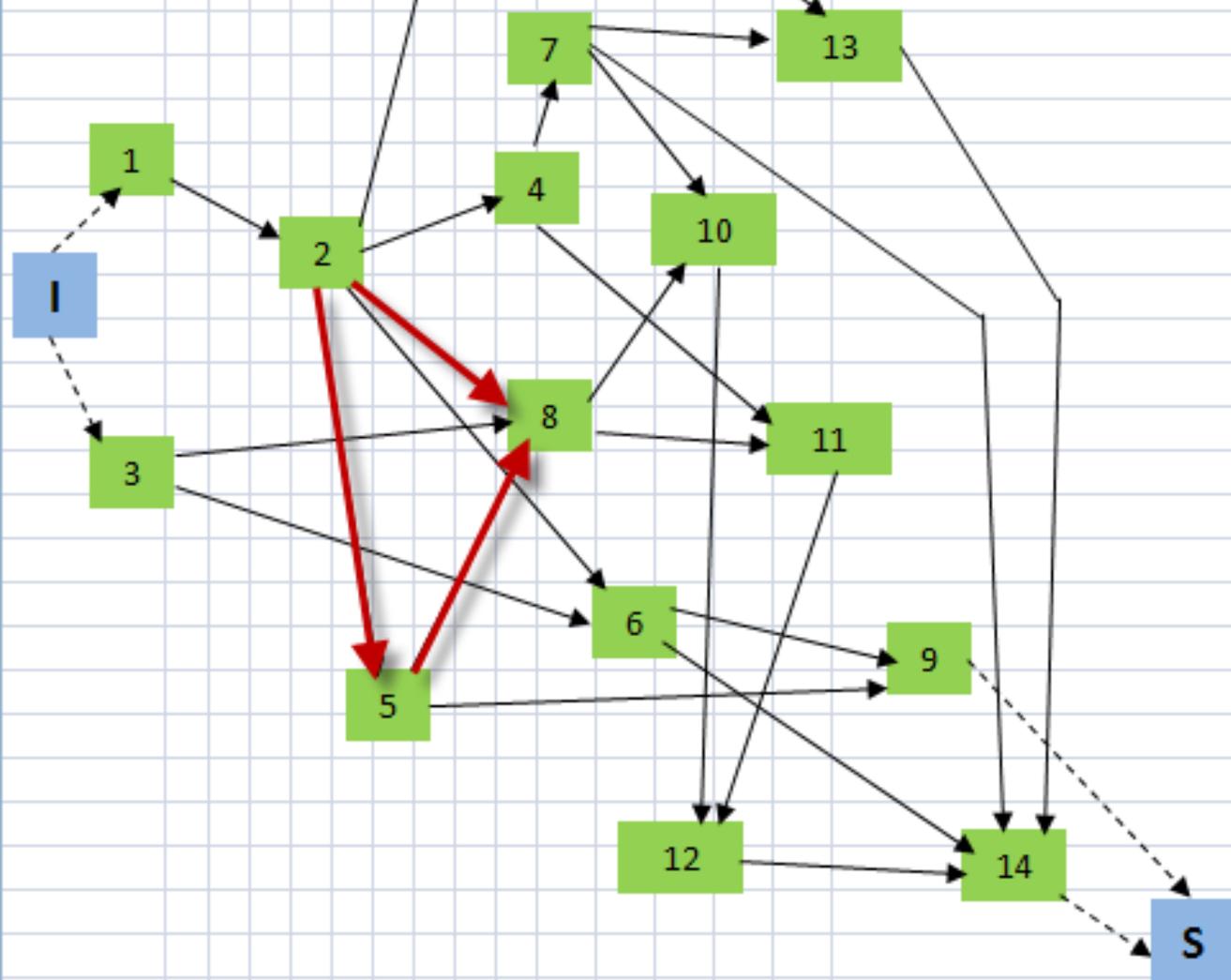
Действие В3: Провести экспертизу по устранению обнаруженного контура и произвести корректировку списка работ сетевой модели (табл.1).

После внесения изменений в список задач $U \rightarrow U'$ (табл.1) вернуться к решению задачи А на скорректированном графе проекта $G'=(J,U')$

Для примера: экспертиза выявила логическую ошибку: Работу (8,2) следует заменить на работу (2,8). Изменения следует внести в список работ и перейти к решению задачи А.

СРСЗ: ЗАДАЧА «А»

(внесена корректировка, работа (2,8)!!!)



CPC3: That's all !



СРС4: ВРЕМЕННЫЕ ПАРАМЕТРЫ ПРОЕКТА

- Определение продолжительностей задач (СРС4.1)
- Расчет главного параметра проекта – критический путь (СРС4.1)
- Расчет временных параметров событий (СРС4.2)
- Расчет временных параметров задач (СРС4.3)

СРС4.1: ДЛИТЕЛЬНОСТИ ЗАДАЧ

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Р	Т	Ь	Т	И	В	Р	Г	И	К	Е	К	–	–
2	О	Р	К	Р	Е	А	Р	В	И	Ь	И	Е	1	1
3	А	И	Г	Ь	Т	О	Е	В	Р	В	Ь	В	1, 2	2
4	В	Р	Р	О	Т	Ь	Ь	О	Ь	Р	Ь	Г	–	–
5	О	О	Р	К	Р	Р	Г	Р	О	Р	И	О	1, 2	2
6	Г	Ч	И	О	А	И	А	Г	А	Р	Р	К	3, 4, 5	3, 4, 5
7	И	В	Г	Т	Р	Т	И	Т	Г	Ч	И	И	1	1
8	Ь	И	О	Г	О	Р	О	Р	И	В	Р	Е	2, 3, 5, 7	3, 5, 7
9	Р	Р	Р	Р	Г	Р	О	В	Ь	Г	Ь	Р	7	7
10	Ч	О	Ч	Р	Ч	Т	Ь	Р	Р	Р	Ч	А	2	2
11	Т	И	О	И	К	Р	А	О	О	И	Т	Ь	2, 3, 4, 7, 8, 9, 10	4, 8, 9, 10
12	В	Т	В	И	О	Т	Т	Г	В	О	Р	В	4, 5, 10	4, 5, 10

- t_{ij} – длительность задачи (I,j) в ед. времени: мин., часы, **дни**, мес., ...

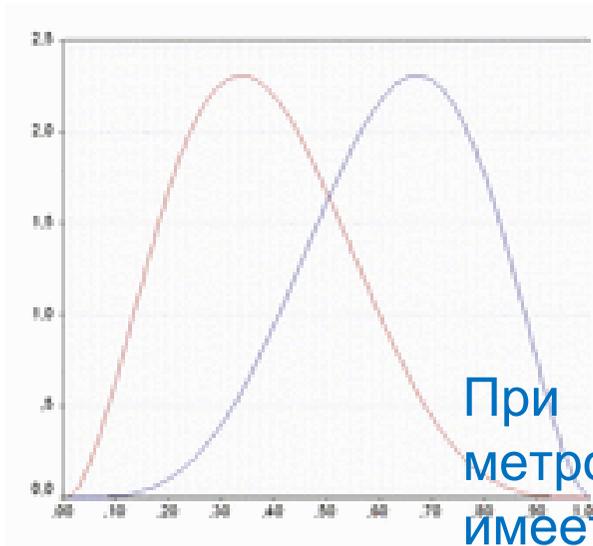
ВАРИАНТЫ ЗАДАНИЯ ДЛИТЕЛЬНОСТИ ЗАДАЧ

СРС4.1: Задание параметров β - РАСПРЕДЕЛЕНИЯ

Работы / Параметры	1	2	3	4	5	6	7	8	9	10	11	12
a	P/18	K/12	A/1	B/3	O/16	G/4	B/3	I/10	P/18	O/16	I/10	B/3
m	T/20	O/16	G/4	P/18	O/16	I/10	G/4	O/16	P/18	Ч/24	O/16	B/3
b	Ь/30	P/18	I/10	P/18	P/18	Ч/24	I/10	Ь/30	P/18	Ч/24	T/20	T/20
t_{ij}												

Алфавит русский нумерованный (пронумерованный) по порядку.

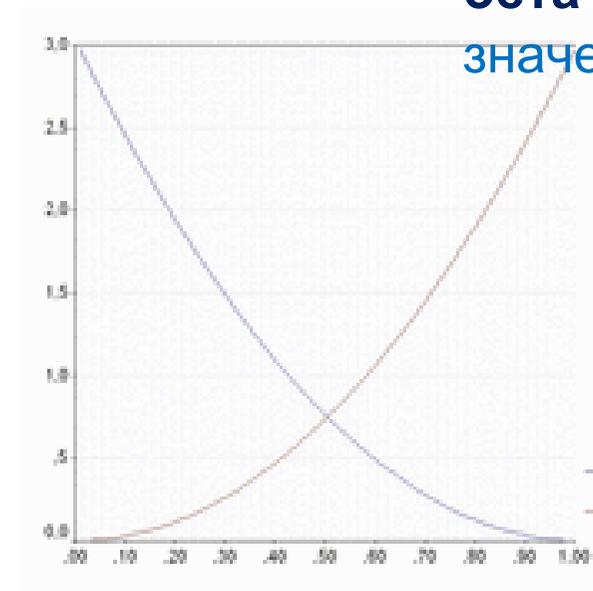
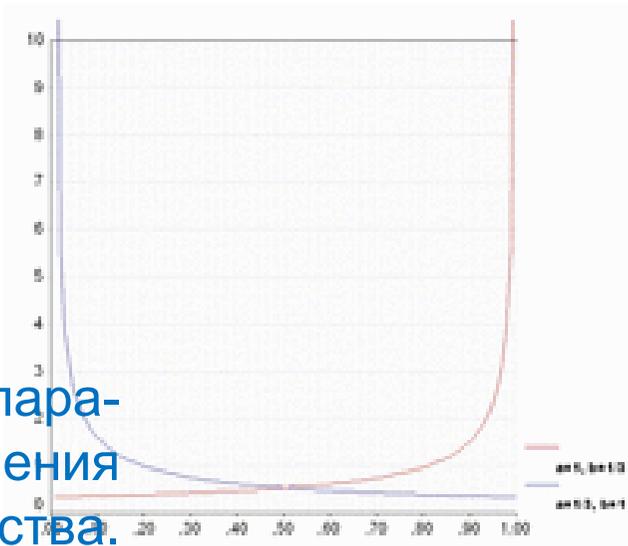
А 1	Б 2	В 3	Г 4	Д 5	Е 6	Ё 7
Ж 8	З 9	И 10	Й 11	К 12	Л 13	М 14
Н 15	О 16	П 17	Р 18	С 19	Т 20	У 21
Ф 22	Х 23	Ц 24	Ч 25	Ш 26	Щ 27	Ъ 28
Ы 29	Ь 30	Э 31	Ю 32	Я 33		



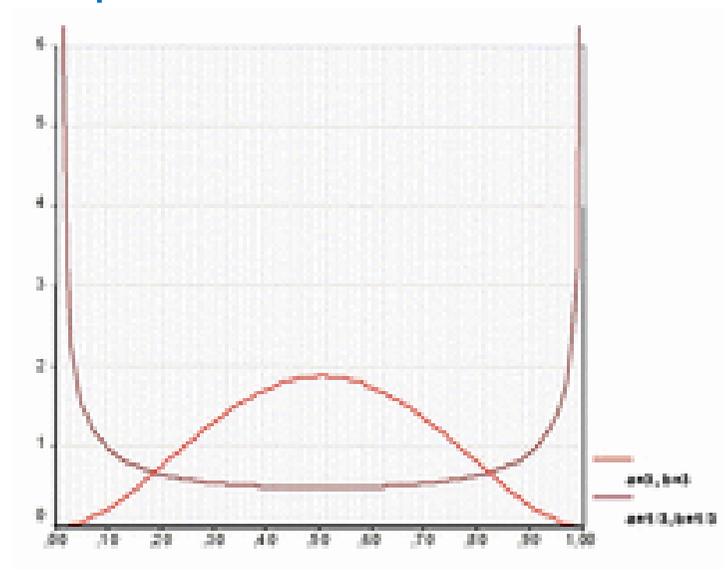
- 1) $a=3, b=5$
 $a=5, b=3$
- 2) $a=1, b=3$
 $a=3, b=1$

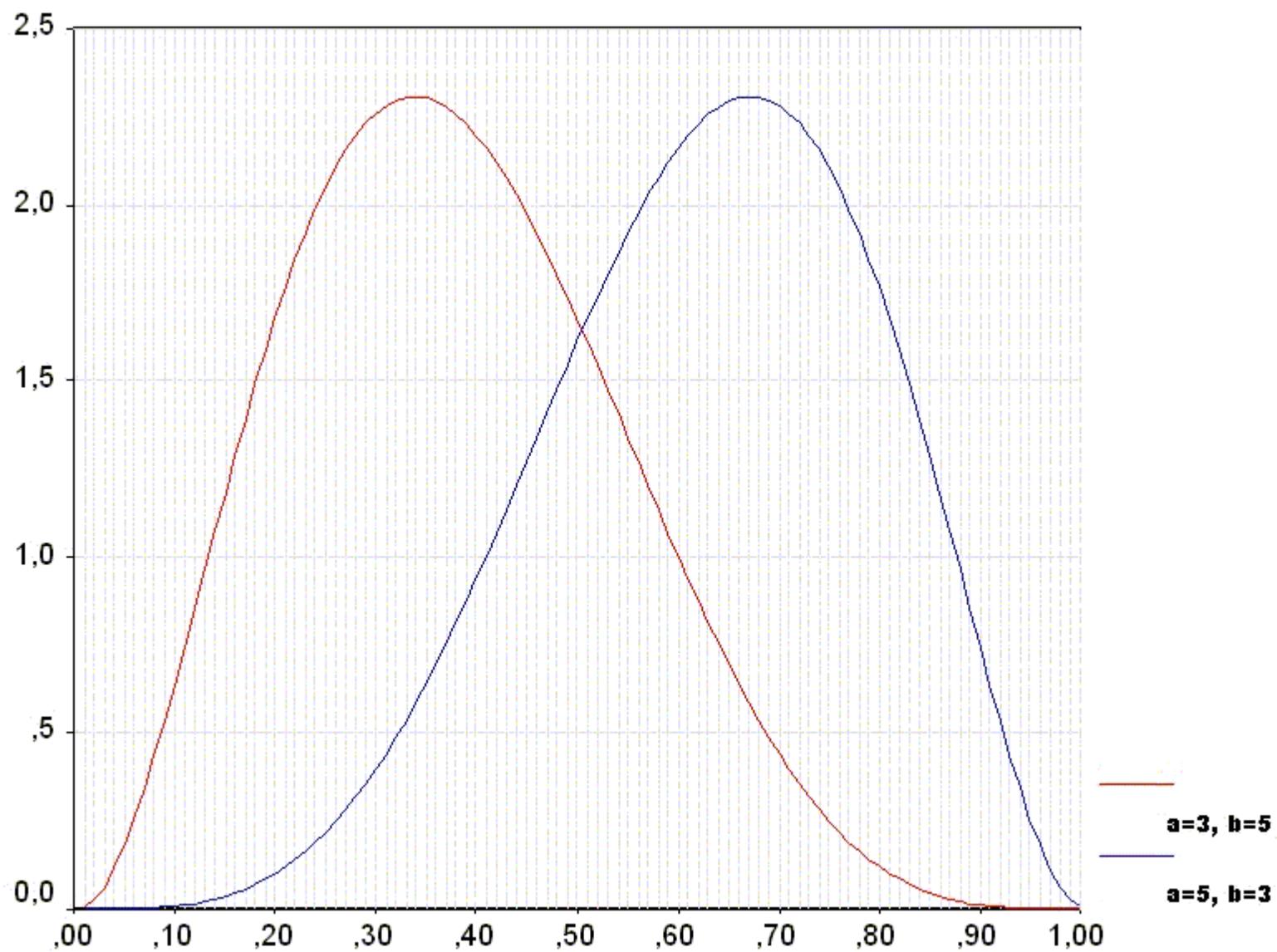
При разных значениях параметров функция распределения имеет самые разные свойства.

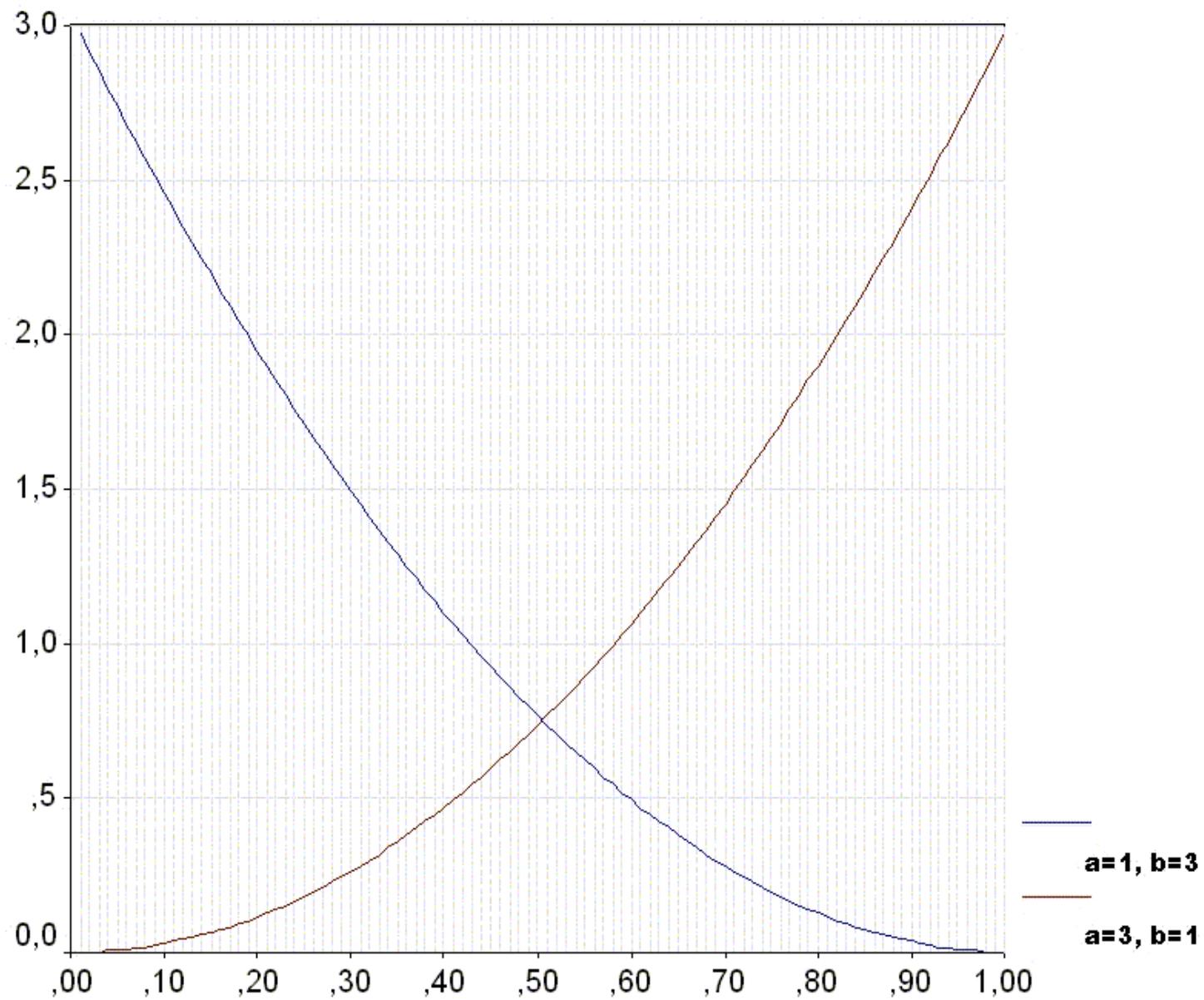
Приведены графики плотности бета-распределения при значениях параметров

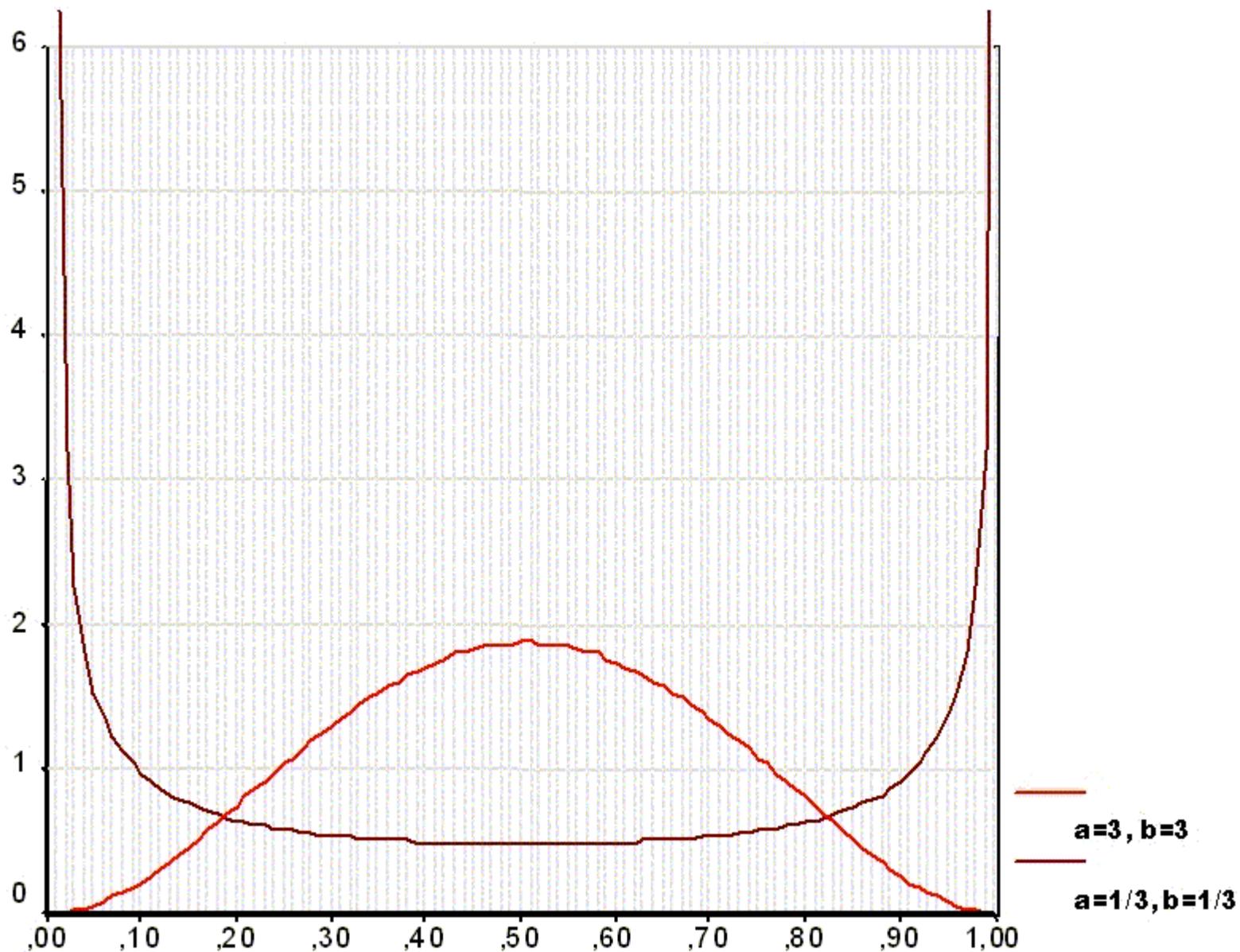


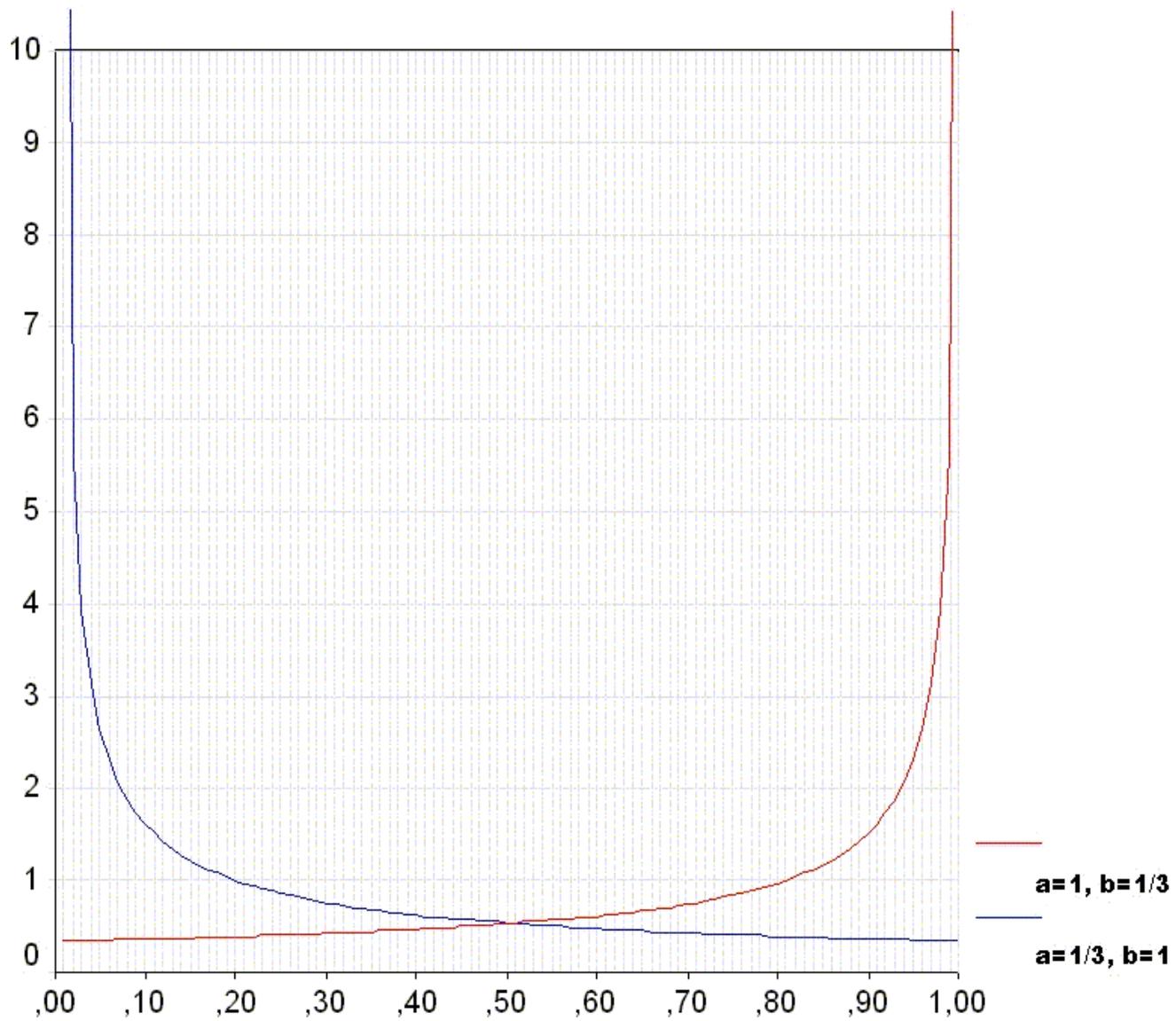
- 3) $a=3, b=3$
 $a=1/3, b=1/3$
- 4) $a=1, b=1/3$
 $a=1/3, b=1$











СРС4.1: β - РАСПРЕДЕЛЕНИЕ

•

1	2	3	4	5	6	7	8	9	10	11	12	
a												
m												
b												
t_{ij}												