Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа компьютерных технологий и информационных систем

Отчёт расчетной работе № 1

Дисциплина: Системный анализ и принятие решений.

Выполнил студент гр. 5130901/10101 ________________ Фоничев И.Р.
(подпись)

Руководитель ________________ Сиднев А.Г.
(подпись)

“__” февраля 2024 г.

Санкт-Петербург
2024

1. Условие:

1.1. Вариант:

Вариант №13.
Граф №13.

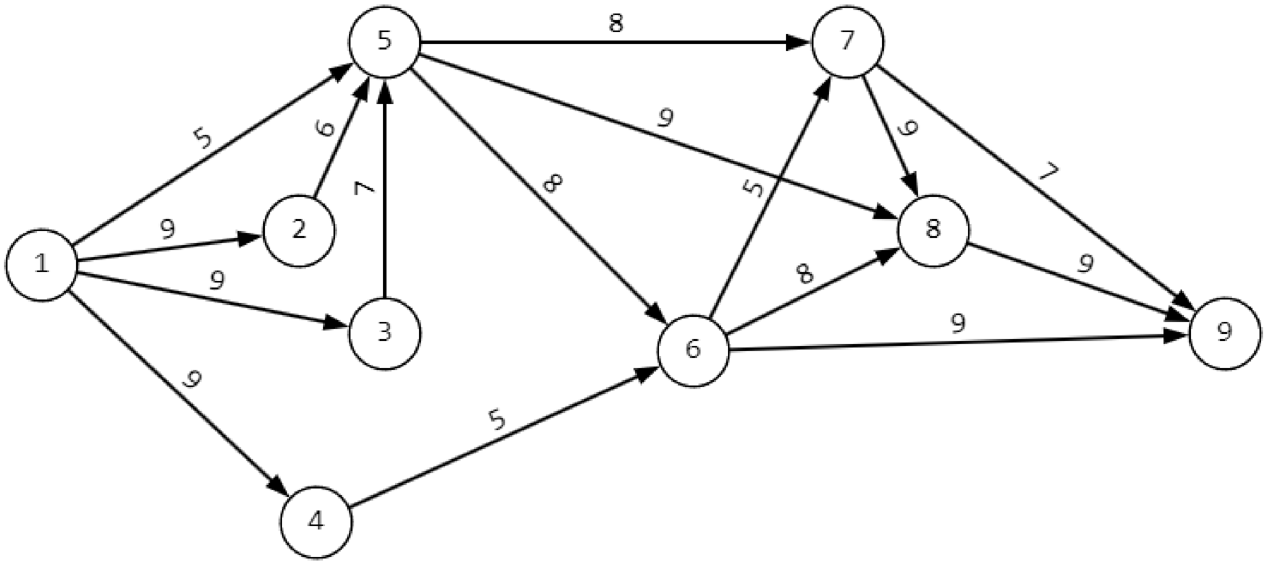

Рис. 1.1. Граф №13.

Число исполнителей 3.
Решающее правило: Короткие работы вперед.

1.2. Условие задания:

Выполнить следующие разделы:

1. Определить наиболее ранние моменты начала работ с использованием метода математического программирования.
2. Определить наиболее ранние моменты начала работ и их интенсивности, если длительность равна интенсивности выполнения работ, а суммарная интенсивность не превышает 75% от общего числа выполняемых работ.
3. Самостоятельно распределить работы между заданным числом исполнителей и сформулировать задачу математического программирования с бинарными индикаторными переменными. Определить число бинарных переменных и дополнительных ограничений в этой задаче и дать содержательную формулировку части ограничений с бинарными переменными.
   3.1. Изменить формулировку задачи так, чтобы число бинарных переменных не превышало 10. Решить полученную задачу с использованием команды **intlinprog**. Определить мощность множества бинарных переменных задачи и дать содержательную интерпретацию полученному решению.
4. Найти характеристики t_i^*, t_i^{**} и r_{ij} расписания выполнения комплекса работ с использованием метода динамического программирования. Привести соответствующие уравнения Беллмана. Определить критические пути на графе.
5. Найти те же характеристики t_i^*, t_i^{**} и r_{ij} расписания выполнения комплекса работ с использованием математического программирования.
6. Определить помимо полных резервов времени $F_n = r_{ij}$ работ ij резервы времени, относящиеся к событиям j сетевого графа, а именно $F_{\text{нз}1}, F_c, F_{\text{нз}2}$.
7. Рассмотреть вероятностную постановку задачи анализа расписания.
   Считать СКО времен выполнения работ равными 5% от их длительностей. Предполагая неизменным критический путь (оценить справедливость этого предположения) найти вероятность того, что время выполнения комплекса работ не превысит найденного для детерминированной задачи в п.1 на 10%.

8. Представить пошаговую процедуру имитационного моделирования расписания по схеме событий с учетом числа исполнителей и решающего правила ранжирования работ из числа возможных. По результатам моделирования построить диаграмму Гантта.

2. Ход решения

2.1. Определить наиболее ранние моменты начала работ с использованием метода математического программирования:

Для графа, представленного на Рис. 1.1, составим систему неравенств для последующего решения с помощью методов линейного программирования. Обозначим за t_{ij} наиболее ранний момент начала работы ij, а за t_{end} – наиболее ранний момент окончания всех работ.

$t_{25} \geq t_{12} + 9$

$t_{35} \geq t_{13} + 9$

$t_{46} \geq t_{14} + 9$

$t_{56} \geq t_{15} + 5$

$t_{56} \geq t_{25} + 6$

$t_{56} \geq t_{35} + 7$

$t_{57} \geq t_{15} + 5$

$t_{57} \geq t_{25} + 6$

$t_{57} \geq t_{35} + 7$

$t_{67} \geq t_{56} + 8$

$t_{67} \geq t_{46} + 5$

$t_{58} \geq t_{15} + 5$

$t_{58} \geq t_{25} + 6$

$t_{58} \geq t_{35} + 7$

$t_{68} \geq t_{46} + 5$

$t_{68} \geq t_{56} + 8$

$t_{78} \geq t_{57} + 8$

$t_{78} \geq t_{67} + 5$

$t_{69} \geq t_{46} + 5$

$t_{69} \geq t_{56} + 8$

$t_{79} \geq t_{57} + 8$

$t_{79} \geq t_{67} + 5$

$t_{89} \geq t_{58} + 9$

$t_{89} \geq t_{68} + 8$

$t_{89} \geq t_{78} + 9$

Задача оптимизации – минимизация следующей функции:

$$min(\sum_{i,j} t_{i,j} + t_{end})$$

Решим эту задачу с помощью функции Matlab linprog. Для этого преобразуем полученные ранее ограничения в матрицы A и b:

```
%   t12 t13 t14 t15 t25 t35 t46 t56 t57 t58 t67 t68 t69 t78 t79 t89
A = [1   0   0   0  -1   0   0   0   0   0   0   0   0   0   0   0; % 1
     0   1   0   0   0  -1   0   0   0   0   0   0   0   0   0   0; % 2
     0   0   1   0   0   0  -1   0   0   0   0   0   0   0   0   0; % 3
     0   0   0   1   0   0   0  -1   0   0   0   0   0   0   0   0; % 4
     0   0   0   0   1   0   0  -1   0   0   0   0   0   0   0   0; % 5
     0   0   0   0   0   1   0  -1   0   0   0   0   0   0   0   0; % 6
     0   0   0   1   0   0   0   0  -1   0   0   0   0   0   0   0; % 7
     0   0   0   0   1   0   0   0  -1   0   0   0   0   0   0   0; % 8
     0   0   0   0   0   1   0   0  -1   0   0   0   0   0   0   0; % 9
     0   0   0   0   0   0   0   1   0   0  -1   0   0   0   0   0; % 10
     0   0   0   0   0   0   1   0   0   0  -1   0   0   0   0   0; % 11
     0   0   0   1   0   0   0   0   0  -1   0   0   0   0   0   0; % 12
     0   0   0   0   1   0   0   0   0  -1   0   0   0   0   0   0; % 13
     0   0   0   0   0   1   0   0   0  -1   0   0   0   0   0   0; % 14
     0   0   0   0   0   0   1   0   0   0   0  -1   0   0   0   0; % 15
     0   0   0   0   0   0   0   1   0   0   0  -1   0   0   0   0; % 16
     0   0   0   0   0   0   0   0   1   0   0   0   0  -1   0   0; % 17
     0   0   0   0   0   0   0   0   0   0   1   0   0  -1   0   0; % 18
     0   0   0   0   0   0   1   0   0   0   0   0  -1   0   0   0; % 19
     0   0   0   0   0   0   0   1   0   0   0   0  -1   0   0   0; % 20
     0   0   0   0   0   0   0   0   1   0   0   0   0   0  -1   0; % 21
     0   0   0   0   0   0   0   0   0   0   1   0   0   0  -1   0; % 22
     0   0   0   0   0   0   0   0   0   1   0   0   0   0   0  -1; % 23
     0   0   0   0   0   0   0   0   0   0   0   1   0   0   0  -1; % 24
     0   0   0   0   0   0   0   0   0   0   0   0   0   1   0  -1; % 25
     ];

b = -[9; 9; 9; 5; 6; 7; 5; 6; 7; 8; 5; 5; 6; 7; 5; 8; 8; 5; 5; 8; 8; 5; 9; 8; 9];
```

После чего вызовем linprog:

```
f = ones(16, 1);
lb = zeros(16, 1);

linprog(f, A, b, [], [], lb, []);
```

Полученный результат выглядит следующим образом:

t_{12}	t_{13}	t_{14}	t_{15}	t_{25}	t_{35}	t_{46}	t_{56}	t_{57}	t_{58}	t_{67}	t_{68}	t_{69}	t_{78}	t_{79}	t_{89}
0	0	0	0	9	9	9	16	16	16	24	24	24	29	29	38

Табл. 2.1. Время начала всех работ.

Теперь мы знаем минимальное время начала каждой работы. Для получения информации о времени выполнения всех работ необходимо к времени начала работы t_{89} прибавить время её выполнения т.е. 9.

Итого время выполнения всех работ равно 47.

2.2. Определить наиболее ранние моменты начала работ и их интенсивности, если длительность равна интенсивности выполнения работ, а суммарная интенсивность не превышает 75% от общего числа выполняемых работ.

Мы сможем увеличить время выполнения всех работ за счет добавления интенсивностей работ, отличных от 1 – некоторые работы ускорим (интенсивность > 1), а некоторые замедлим (интенсивность < 1), если это потребуется.

Изменим исходную систему неравенств согласно правилу:

$$\min\left\{\sum_{i,j} t_{ij}\right\}$$

$$\begin{cases} t_{ij} \geq \tau_{li} + \dfrac{Q_{li}}{m_{li}}, i = \overline{1, M-1}; l \in G^{-}(i) \\ \sum\limits_{i,j} m_{ij} \leq 0{,}75 * 15, l \in G^{-}(M) \\ t_{ij} \geq 0 \end{cases}$$

где m_{ij} – интенсивность ij работы.

Такими условиями мы пытаемся минимизировать время начала всех работ, при интенсивности, не превышающей 75% от числа выполняемых работ т.е. 15.

Создадим набор всех «работ» т.е. ребер графа и массив троек, где закодируем систему неравенств, созданную ранее:

```
function [] = task_2()
    works = [12 13 14 15 25 35 46 56 57 58 67 68 69 78 79 89 99];
    conds = [
        25 12 9
        35 13 9
        46 14 9
        56 15 5
        56 25 6
        56 35 7
        57 15 5
        57 25 6
        57 35 7
        67 56 8
        67 46 5
        58 15 5
        58 25 6
        58 35 7
        68 46 5
        68 56 8
        78 57 8
        78 67 5
        69 46 5
        69 56 8
        79 57 8
        79 67 5
        89 58 9
        89 68 8
        89 78 9
        99 69 9
        99 79 7
        99 89 9
        ];
```

Стоит заметить, что появилась работа-фальшивка. Это необходимо, чтоб MATLAB оптимизировал также и путь из 8 в 9 вершину и выводил нам результат этой оптимизации.

Создадим необходимые параметры для fmincon, а также функцию, которая распарсит заданные нам тройки в требуемые для fmincon значения и выведем результат выполнения на экран:

```
    x0 = ones(length(works) * 2 - 1, 1);
    lb = zeros(length(works) * 2 - 1, 1);

    fun = @(x) sum(x(1:length(works)));

    res = fmincon(fun, x0, [], [], [], [], lb, [], @funs);

    function [c, ceq] = funs(x)
        c = [];
        for i = 1:length(conds)
            t1 = find(works == conds(i, 1));
            t2 = find(works == conds(i, 2));
            q = conds(i, 3);
            m = length(works) + t2;
            c(end + 1) = -x(t1(1)) + x(t2(1)) + q / x(m);
        end
        ceq = sum(x(length(works) + 1:end)) - 0.75 * (length(works) - 1);
    end

    t_res = res(1:length(works));
    m_res = res(length(works) + 1:end);
    sum_m_res = sum(m_res);
    mean_m_res = mean(m_res);

    disp(t_res);
    disp(m_res);
    disp(sum_m_res);
    disp(mean_m_res);
end
```

В результате выполнения получим следующие значения:

Моменты начала работ	
t_{12}	0.0000
t_{13}	0.0000
t_{14}	0.0000
t_{15}	0.0000
t_{25}	6.3451
t_{35}	6.1364
t_{46}	9.2213
t_{56}	12.1638
t_{57}	12.1638
t_{58}	12.1638
t_{67}	18.3867
t_{68}	18.3867
t_{69}	18.3867
t_{78}	25.3845
t_{79}	25.3845
t_{89}	39.1247
t_{end}	55.8086

Интенсивности	
m_{12}	1.4184
m_{13}	1.4667
m_{14}	0.9760
m_{15}	0.4111
m_{25}	1.0312
m_{35}	1.1614
m_{46}	0.5455
m_{56}	1.2856
m_{57}	0.6051
m_{58}	0.3338
m_{67}	0.7145
m_{68}	0.3858
m_{69}	0.2405
m_{78}	0.6550
m_{79}	0.2301
m_{89}	0.5394

Табл. 2.2. Результат выполнения программы.

Сумма интенсивностей равна 12, что составляет ровно 75% от числа исполняемых работ, как и требовалось в задании. Стоит отметить, что общее время работы возрастало до 55.8 с 47 т.е. на 8.8 секунд. Это связано с тем, что при уменьшении интенсивности, некоторые работы стали работать дольше.

2.3. Самостоятельно распределить работы между заданным числом исполнителей и сформулировать задачу математического программирования с бинарными индикаторными переменными. Определить число бинарных переменных и дополнительных ограничений в этой задаче и дать содержательную формулировку части ограничений с бинарными переменными.

Распределим 16 работ по трем исполнителям следующим образом:

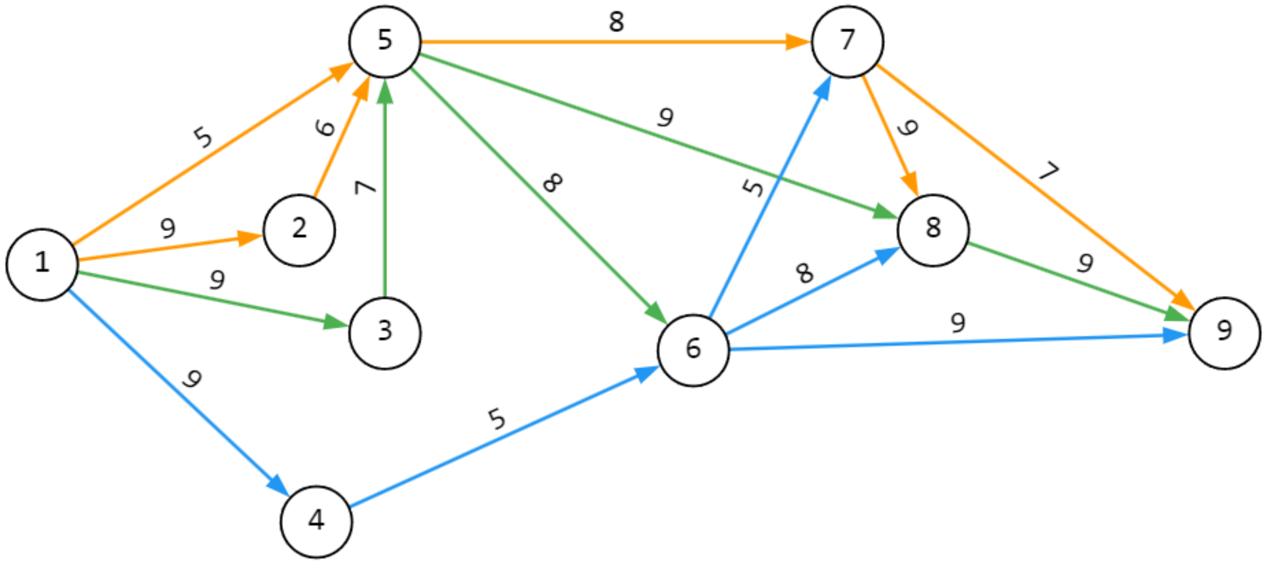

Рис. 2.1. Граф с задачами, распределенными по исполнителям.

Таким образом желтый цвет – первый исполнитель (6 задач), зеленый цвет – второй исполнитель (5 задач), а синий – третий (5 задач).

Составим следующую систему для каждой пары работ $\{ij, lm\}$:

$$\begin{cases} (M+\tau_{lm})Y_{ij,lm,k} + (t_{ij} - t_{lm}) \geq \tau_{lm}, \\ (M+\tau_{ij})Y_{lm,ij,k} + (t_{lm} - t_{ij}) \geq \tau_{ij}, \\ Y_{ij,lm,k} + Y_{lm,ij,k} = 1 \end{cases}$$

где $|M| \gg \sum_{\{ij\}} \tau_{ij}$, тогда число дополнительных ограничений задачи с бинарными переменными будет равно $3(C_6^2 + C_5^2 + C_5^2) = 3(15 + 10 + 10) = 105$, а число бинарных переменных $2(C_6^2 + C_5^2 + C_5^2) = 2(15 + 10 + 10) = 70$.

2.3.1. Изменить формулировку задачи так, чтобы число бинарных переменных не превышало 10. Решить полученную задачу с использованием команды intlinprog. Определить мощность множества бинарных переменных задачи и дать содержательную интерпретацию полученному решению.

Упростим поставленную задачу, пусть только некоторые задачи выполняются определенным исполнителем, а над остальными может работать неограниченное число исполнителей:

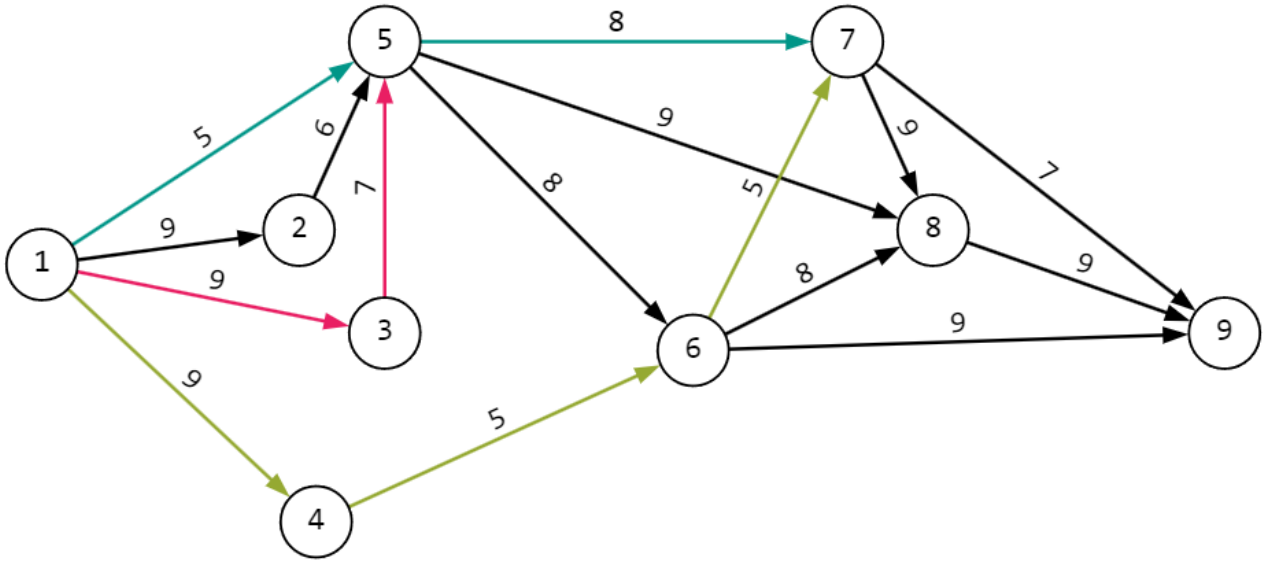

Рис. 2.2. Граф с задачами, часть которых распределена по исполнителям.

Таким образом синий цвет – первый исполнитель (2 задачи), розовый цвет – второй исполнитель (2 задачи), а зеленый – третий (3 задачи), что дает нам $2(C_2^2 + C_2^2 + C_3^2) = 2(1+1+3) = 10$ дополнительных бинарных переменных и $3(C_2^2 + C_2^2 + C_3^2) = 3(1+1+3) = 15$ дополнительных ограничений:

$$\begin{cases} (M+\tau_{13})Y_{35,13,1} + (t_{35} - t_{13}) \geq \tau_{13}, \\ (M+\tau_{35})Y_{13,35,1} + (t_{13} - t_{35}) \geq \tau_{35}, \\ Y_{35,13,1} + Y_{13,35,1} = 1 \end{cases}$$

$$\begin{cases} (M+\tau_{57})Y_{15,57,2} + (t_{15} - t_{57}) \geq \tau_{57}, \\ (M+\tau_{15})Y_{57,15,2} + (t_{57} - t_{15}) \geq \tau_{15}, \\ Y_{15,57,2} + Y_{57,15,2} = 1 \end{cases}$$

$$\begin{cases} (M+\tau_{14})Y_{46,14,3} + (t_{46} - t_{14}) \geq \tau_{14}, \\ (M+\tau_{46})Y_{14,46,3} + (t_{14} - t_{46}) \geq \tau_{46}, \\ Y_{46,14,3} + Y_{14,46,3} = 1 \end{cases}$$

$$\begin{cases} (M+\tau_{46})Y_{67,46,4} + (t_{67} - t_{46}) \geq \tau_{46}, \\ (M+\tau_{67})Y_{46,67,4} + (t_{46} - t_{67}) \geq \tau_{67}, \\ Y_{67,46,4} + Y_{46,67,4} = 1 \end{cases}$$

После чего решим задачу Matlab с использованием функции intlinprog. Для начала зададим наш граф, как делали это ранее, также добавим пары для новых ограничений и длины путей на каждой из задач:

```
function [] = task_3_1()
    works = [12 13 14 15 25 35 46 56 57 58 67 68 69 78 79 89 99];
    conds = [
        25 12 9
        35 13 9
        46 14 9
        56 15 5
        56 25 6
        56 35 7
        57 15 5
        57 25 6
        57 35 7
        67 56 8
        67 46 5
        58 15 5
        58 25 6
        58 35 7
        68 46 5
        68 56 8
        78 57 8
        78 67 5
        69 46 5
        69 56 8
        79 57 8
        79 67 5
        89 58 9
        89 68 8
        89 78 9
        99 69 9
        99 79 7
        99 89 9
        ];

    pairs = [
        15 57 5 8
        13 35 9 7
        14 46 9 5
        46 67 5 5
    ];
```

Далее создадим массив, как в пункте 2.1, но используя наши объявления:

```
cond_len = length(conds) + 2 * length(pairs);
x_len = length(works) + 2 * length(pairs);
f = ones(x_len, 1);
f(length(works) + 1:end) = 0;
A = zeros(cond_len, x_len);
b = zeros(1, cond_len);
for i = 1:length(conds)
    t1 = find(works == conds(i, 1));
    t2 = find(works == conds(i, 2));
    A(i, t1) = -1;
    A(i, t2) = 1;
    b(1, i) = -conds(i, 3);
end
```

Теперь необходимо создадим уравнения, которые добавились дополнительными ограничениями:

```
Aeq = zeros(length(pairs), x_len);
beq = ones(1, length(pairs));

for i = 1:length(pairs)
    t1 = find(works == pairs(i, 1));
    t2 = find(works == pairs(i, 2));
    tau1 = pairs(i, 3);
    tau2 = pairs(i, 4);
    Y1 = length(works) + 2 * i - 1;
    Y2 = length(works) + 2 * i;
    M = 1000;

    idx = length(conds) + 2 * i - 1;
    A(idx, Y1) = -(M + tau2);
    A(idx, t1) = -1;
    A(idx, t2) = 1;
    b(1, idx) = -tau2;

    idx = idx + 1;
    A(idx, Y2) = -(M + tau1);
    A(idx, t1) = 1;
    A(idx, t2) = -1;
    b(1, idx) = -tau1;

    Aeq (i, Y1) = 1;
    Aeq (i, Y2) = 1;
end
```

Все необходимые переменные были созданы, теперь запустим intlinprog и посмотрим на результат:

```
lb = zeros(x_len, 1);

x = intlinprog(f, (length(works) + 1):x_len, A, b, Aeq, beq, lb);

disp(x)
```

В результате запуска получим следующие значения:

Моменты начала работ	
t_{12}	0
t_{13}	0
t_{14}	0
t_{15}	0
t_{25}	9
t_{35}	9
t_{46}	9
t_{56}	16
t_{57}	16
t_{58}	16
t_{67}	24
t_{68}	24
t_{69}	24
t_{78}	29
t_{79}	29
t_{89}	38
t_{end}	47

Бинарные переменные	
$Y_{15,57,1}$	1
$Y_{57,15,1}$	0
$Y_{13,35,2}$	1
$Y_{35,13,2}$	0
$Y_{14,46,3}$	1
$Y_{46,14,3}$	0
$Y_{46,67,4}$	1
$Y_{67,46,4}$	0

Табл. 2.3. Результат работы программы.

Как мы видим по значениям бинарных переменных и времени начала работы наши условия выполняются.

Первый исполнитель выполняет сначала работу 15, а потом 57 (т. к. значение t_{15}= 0, а t_{57}= 16)

2.4. Найти характеристики t_i^*, t_i^{**} и r_{ij} расписания выполнения комплекса работ с использованием метода динамического программирования. Привести соответствующие уравнения Беллмана. Определить критические пути на графе.

Каждому узлу на графе можно сопоставить два момента: минимальное время, когда событие будет осуществлено t_i^* и наиболее поздний момент t_i^{**}.

Воспользуемся методом динамического программирования и определим наиболее ранние моменты t_i^* для каждого узла графа:

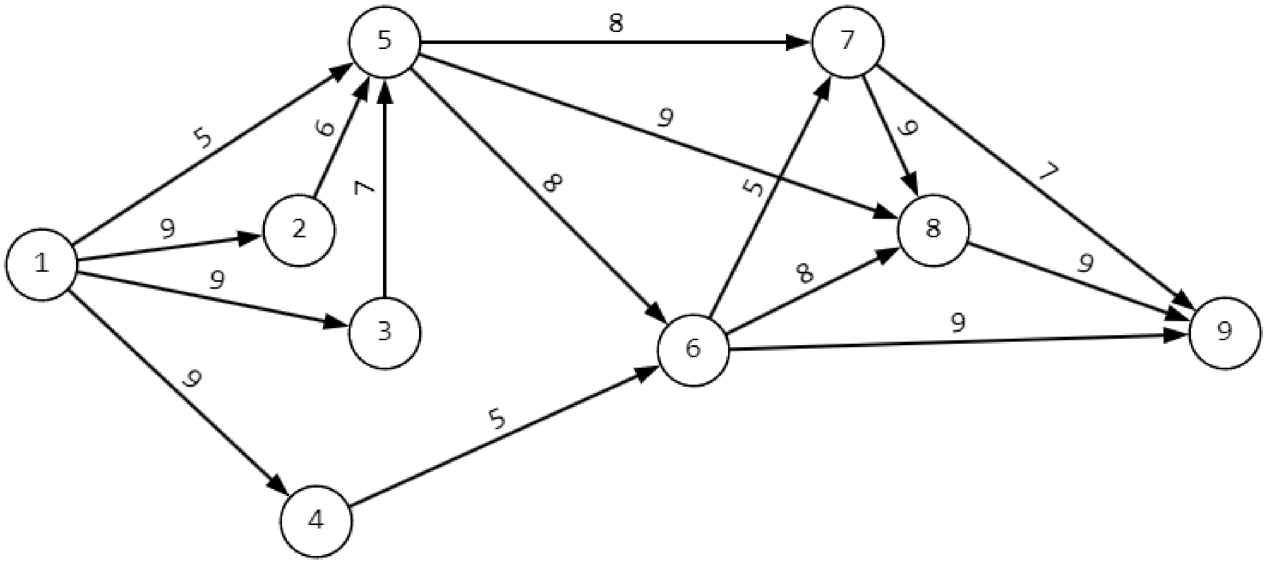

Рис. 2.3. Граф.

Самое ранее время начала выполнения работы – время, когда выполнятся все работы, предшествующие заданной:

$t_1^* = 0$

$t_2^* = t_1^* + \tau_{12} = 9$

$t_3^* = t_1^* + \tau_{13} = 9$

$t_4^* = t_1^* + \tau_{14} = 9$

$t_5^* = max\{t_1^* + \tau_{15};\ t_2^* + \tau_{25};\ t_3^* + \tau_{35}\} = max\{5; 15;\ 16\} = 16$

$t_6^* = max\{t_4^* + \tau_{46};\ t_5^* + \tau_{56}\} = max\{14;\ 24\} = 24$

$t_7^* = max\{t_5^* + \tau_{57};\ t_6^* + \tau_{67}\} = max\{24; 29\} = 29$

$t_8^* = max\{t_7^* + \tau_{78};\ t_6^* + \tau_{68};\ t_5^* + \tau_{58}\} = max\{38; 32; 25\} = 38$

$t_9^* = max\{t_7^* + \tau_{79};\ t_6^* + \tau_{69};\ t_8^* + \tau_{89}\} = max\{36; 33; 48\} = 47$

Полученные значение совпадают с полученными в предыдущих пунктах, что свидетельствует о корректности проделанной работы.

Теперь используя полученные значения определим наиболее поздние моменты времени t_i^{**}:

$t_9^{**} = 47$

$t_8^{**} = \min\{t_9^{**} - \tau_{89};\ t_9^{**} - \tau_{79};\ t_9^{**} - \tau_{69}\} = \min\{38; 41; 38\} = 38$

$t_7^{**} = \min\{t_9^{**} - \tau_{79};\ t_8^{**} - \tau_{78}\} = \min\{40; 29\} = 29$

$t_6^{**} = \min\{t_7^{**} - \tau_{67};\ t_8^{**} - \tau_{68};\ t_9^{**} - \tau_{69}\} = \min\{22; 30; 38\} = 24$

$t_5^{**} = \min\{t_7^{**} - \tau_{57};\ t_8^{**} - \tau_{58};\ t_6^{**} - \tau_{56}\} = \min\{21; 29; 14\} = 16$

$t_4^{**} = t_6^{**} - \tau_{46} = 24 - 5 = 19$

$t_3^{**} = t_5^{**} - \tau_{35} = 16 - 7 = 9$

$t_2^{**} = t_5^{**} - \tau_{25} = 16 - 6 = 10$

$t_1^{**} = \min\{t_2^{**} - \tau_{12};\ t_3^{**} - \tau_{13};\ t_4^{**} - \tau_{14};\ t_5^{**} - \tau_{15}\} = \min\{1; 0; 10; 11\} = 0$

По формуле $r_{ij} = t_j^{**} - (t_i^* + \tau_{ij})$ определим резервы времени выполнения всех работ:

$r_{12} = t_2^{**} - (t_1^* + \tau_{12}) = 10 - 9 = 1$

$\boldsymbol{r_{13} = t_3^{**} - (t_1^* + \tau_{13}) = 9 - 9 = 0}$

$r_{14} = t_4^{**} - (t_1^* + \tau_{14}) = 19 - 9 = 10$

$r_{15} = t_5^{**} - (t_1^* + \tau_{15}) = 16 - 5 = 11$
$r_{25} = t_5^{**} - (t_2^* + \tau_{25}) = 16 - 16 = 1$
$\boldsymbol{r_{35} = t_5^{**} - (t_3^* + \tau_{35}) = 16 - 16 = 0}$
$r_{46} = t_6^{**} - (t_4^* + \tau_{46}) = 24 - 14 = 10$
$\boldsymbol{r_{56} = t_6^{**} - (t_5^* + \tau_{56}) = 24 - 24 = 0}$
$r_{57} = t_7^{**} - (t_5^* + \tau_{57}) = 29 - 24 = 5$
$r_{58} = t_8^{**} - (t_5^* + \tau_{58}) = 38 - 25 = 13$
$\boldsymbol{r_{67} = t_7^{**} - (t_6^* + \tau_{67}) = 29 - 29 = 0}$
$r_{68} = t_8^{**} - (t_6^* + \tau_{68}) = 38 - 32 = 6$
$r_{69} = t_9^{**} - (t_6^* + \tau_{69}) = 47 - 33 = 14$
$\boldsymbol{r_{78} = t_8^{**} - (t_7^* + \tau_{78}) = 38 - 38 = 0}$
$r_{79} = t_9^{**} - (t_7^* + \tau_{79}) = 47 - 36 = 11$
$\boldsymbol{r_{89} = t_9^{**} - (t_8^* + \tau_{89}) = 47 - 47 = 0}$

Работы, у которых резерв равен 0 – критический путь. Их длительность напрямую влияет на продолжительность выполнения всех работ.

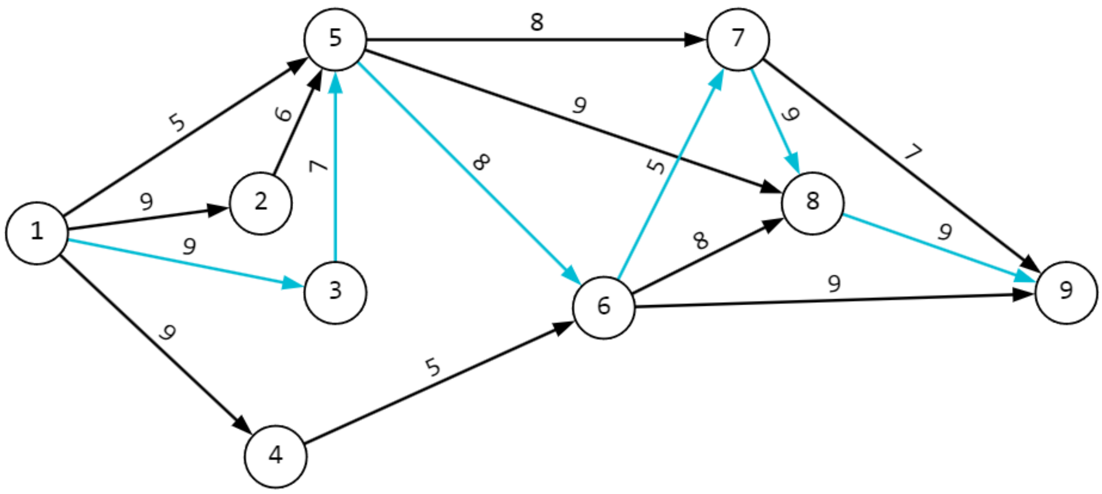

Рис. 2.4. Критический путь в графе.

2.5. Найти те же характеристики t_i^*, t_i^{**} и r_{ij} расписания выполнения комплекса работ с использованием математического программирования.

Оптимизационная задача для поиска наиболее ранних моментов t_i^* может быть сформулирована следующим образом:

$$\min\left\{\sum_{j=1} t_j^*\right\}$$

$$t_i^* - t_j^* \leq -\tau_{ij}, j = \overline{1,n}$$

Для исходного графа получим следующую оптимизационную задачу:

$$\min\left\{\sum_{j=1} t_j^*\right\}$$

$t_1^* - t_2^* \leq -9$	$t_3^* - t_5^* \leq -7$	$t_6^* - t_7^* \leq -5$
$t_1^* - t_3^* \leq -9$	$t_4^* - t_6^* \leq -15$	$t_6^* - t_8^* \leq -14$
$t_1^* - t_4^* \leq -9$	$t_5^* - t_6^* \leq -8$	$t_6^* - t_9^* \leq -23$
$t_1^* - t_5^* \leq -16$	$t_5^* - t_7^* \leq -13$	$t_7^* - t_8^* \leq -9$
$t_2^* - t_5^* \leq -7$	$t_5^* - t_8^* \leq -22$	$t_7^* - t_9^* \leq -18$
		$t_8^* - t_9^* \leq -9$

$$t_j \geq 0, j = \overline{1{,}9}$$

Запишем эти выражения в массивы:

```
function [] = task_5_1()
    times = [1 2 3 4 5 6 7 8 9];
    conds = [
        1 2 -9
        1 3 -9
        1 4 -9
        1 5 -16
        2 5 -7
        3 5 -7
        4 6 -15
        5 6 -8
        5 7 -13
        5 8 -22
        6 7 -5
        6 8 -14
        6 9 -23
        7 8 -9
        7 9 -18
        8 9 -9
    ];
```

Запишем эти выражения в виде, подходящем для linprog и вычислим:

```
    f = ones(length(times), 1);
    A = zeros(length(conds), length(times));
    b = zeros(1, length(conds));
    for i = 1:length(conds)
        t1 = find(times == conds(i, 1));
        t2 = find(times == conds(i, 2));
        A(i, t1) = 1;
        A(i, t2) = -1;
        b(1, i) = conds(i, 3);
    end

    lb = zeros(length(times), 1);

    t = linprog(f, A, b, [], [], lb);
    disp(t)
end
```

Полученный результат выглядит следующим образом:

t_1^*	t_2^*	t_3^*	t_4^*	t_5^*	t_6^*	t_7^*	t_8^*	t_9^*
0	9	9	9	16	24	29	38	47

Табл. 2.4. Результат решения программой.

Как можно заметить, они идентичны полученным ранее другим способом.

Оптимизационная задача для поиска наиболее поздних моментов t_i^{**} может быть сформулирована следующим образом:

$$max\left\{\sum_{j=1} t_j^{**}\right\}$$

$$t_i^{**} - t_j^{**} \leq -\tau_{ij}, j = \overline{1,n}$$

$$-t_1^{**} + t_9^{**} = 47$$

$$t_1^{**} = 0$$

Для решения поставленной задачи необходимо чуть-чуть изменить вызов функции, объявления условий останется аналогичным:

```
    f = ones(length(times), 1);
    A = zeros(length(conds), length(times));
    b = zeros(1, length(conds));
    for i = 1:length(conds)
        t1 = find(times == conds(i, 1));
        t2 = find(times == conds(i, 2));
        A(i, t1) = 1;
        A(i, t2) = -1;
        b(1, i) = conds(i, 3);
    end

    lb = zeros(length(times), 1);
    Aeq = zeros(2, length(times));
    Aeq(1, 1) = 1;
    Aeq(2, 1) = -1;
    Aeq(2, length(times)) = 1;
    beq = [0; 47];

    t = linprog(-f, A, b, Aeq, beq, lb);
    disp(t)
end
```

Результатом выполнения будет следующим:

t_1^{**}	t_2^{**}	t_3^{**}	t_4^{**}	t_5^{**}	t_6^{**}	t_7^{**}	t_8^{**}	t_9^{**}
0	10	9	19	16	24	29	38	47

Табл. 2.5. Результат выполнения программы.

Как можно заметить, эти значения идентичны посчитанным ранее.
Очевидно, что значение r_{ij} будет аналогично равно посчитанному ранее.

2.6. Определить помимо полных резервов времени $F_n = r_{ij}$ работ ij резервы времени, относящиеся к событиям j сетевого графа, а именно $F_{\text{нз1}}, F_{\text{с}}, F_{\text{нз2}}$.

По формуле $F_{j\text{Н31}} = t_j^{**} - t_j^{*}$ определим независимые резервы 1-го порядка

j	1	2	3	4	5	6	7	8	9
t_j^{*}	0	9	9	9	16	24	29	38	47
t_j^{**}	0	10	9	19	16	24	29	38	47
$F_{j\text{Н31}}$	0	1	0	10	0	0	0	0	0

Табл. 2.6. Независимый резерв 1-го порядка.

По формуле $F_{ijC} = t_j^{*} - (t_i^{*} + \tau_{ij})$ определим свободные резервы времени:

$F_{12C} = t_2^{*} - (t_1^{*} + \tau_{12}) = 9 - 9 = 0$
$F_{13C} = t_3^{*} - (t_1^{*} + \tau_{13}) = 9 - 9 = 0$
$F_{14C} = t_4^{*} - (t_1^{*} + \tau_{14}) = 9 - 9 = 0$
$F_{15C} = t_5^{*} - (t_1^{*} + \tau_{15}) = 16 - 5 = 11$
$F_{25C} = t_5^{*} - (t_2^{*} + \tau_{25}) = 16 - 15 = 1$
$F_{35C} = t_5^{*} - (t_3^{*} + \tau_{35}) = 16 - 16 = 0$
$F_{46C} = t_6^{*} - (t_4^{*} + \tau_{46}) = 24 - 14 = 10$
$F_{56C} = t_6^{*} - (t_5^{*} + \tau_{56}) = 24 - 24 = 0$
$F_{57C} = t_7^{*} - (t_5^{*} + \tau_{57}) = 29 - 24 = 5$
$F_{58C} = t_8^{*} - (t_5^{*} + \tau_{58}) = 38 - 25 = 13$
$F_{67C} = t_7^{*} - (t_6^{*} + \tau_{67}) = 29 - 29 = 0$

$F_{68C} = t_8^* - (t_6^* + \tau_{68}) = 38 - 32 = 6$
$F_{69C} = t_9^* - (t_6^* + \tau_{69}) = 47 - 33 = 14$
$F_{78C} = t_8^* - (t_7^* + \tau_{78}) = 38 - 38 = 0$
$F_{79C} = t_9^* - (t_7^* + \tau_{79}) = 47 - 36 = 11$
$F_{89C} = t_9^* - (t_8^* + \tau_{89}) = 47 - 47 = 0$

По формуле $F_{ij\text{Н32}} = t_j^* - (t_i^{**} + \tau_{ij})$ определим независимые резервы 2-го порядка:

$F_{12H32} = t_2^* - (t_1^{**} + \tau_{12}) = 9 - 9 = 0$
$F_{13H32} = t_3^* - (t_1^{**} + \tau_{13}) = 9 - 9 = 0$
$F_{14H32} = t_4^* - (t_1^{**} + \tau_{14}) = 9 - 9 = 0$
$F_{15H32} = t_5^* - (t_1^{**} + \tau_{15}) = 16 - 5 = 11$
$F_{25H32} = t_5^* - (t_2^{**} + \tau_{25}) = 16 - 16 = 0$
$F_{35H32} = t_5^* - (t_3^{**} + \tau_{35}) = 16 - 16 = 0$
$F_{46H32} = t_6^* - (t_4^{**} + \tau_{46}) = 24 - 24 = 0$
$F_{56H32} = t_6^* - (t_5^{**} + \tau_{56}) = 24 - 24 = 0$
$F_{57H32} = t_7^* - (t_5^{**} + \tau_{57}) = 29 - 24 = 5$
$F_{58H32} = t_8^* - (t_5^{**} + \tau_{58}) = 38 - 25 = 13$
$F_{67H32} = t_7^* - (t_6^{**} + \tau_{67}) = 29 - 29 = 0$
$F_{68H32} = t_8^* - (t_6^{**} + \tau_{68}) = 38 - 32 = 6$
$F_{69H32} = t_9^* - (t_6^{**} + \tau_{69}) = 47 - 33 = 14$
$F_{78H32} = t_8^* - (t_7^{**} + \tau_{78}) = 38 - 38 = 0$
$F_{79H32} = t_9^* - (t_7^{**} + \tau_{79}) = 47 - 36 = 11$
$F_{89H32} = t_9^* - (t_8^{**} + \tau_{89}) = 47 - 47 = 0$

2.7. Рассмотреть вероятностную постановку задачи анализа расписания. Считать СКО времен выполнения работ равными 5% от их длительностей. Предполагая неизменным критический путь (оценить справедливость этого предположения) найти вероятность того, что время выполнения комплекса работ не превысит найденного для детерминированной задачи в п.1 на 10%.

Оценим справедливость неизменности критического пути. Среднее значение длительности работ в графе равно 7.625 временных единиц. По условию СКО равно 5%, то есть $7.625 * 0.05 = 0.38125$. Следовательно, значение длительности работы может отклониться более чем на 1 с очень маленькой вероятностью (по правилу трех сигм). Так как минимальные временной резерв у работы, не лежащей на критическом пути равен 1, то вероятность изменения критического пути очень мала.

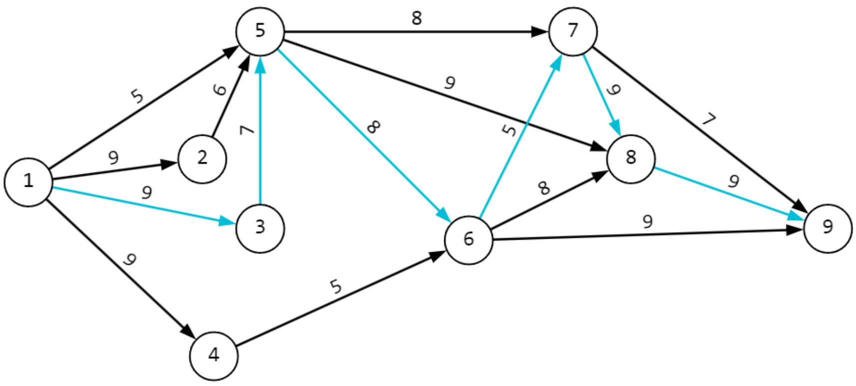

Рис. 2.5. Критический путь в графе.

Математическое ожидание суммы случайных величин равно сумме математических ожиданий:

$$\sum M_{ij} = \sum t_{ij} = 9 + 7 + 8 + 5 + 9 + 9 = 47$$

Дисперсия суммы равна сумме дисперсий:

$$\sum D_{ij} = 0.05^2 * (9^2 + 7^2 + 8^2 + 5^2 + 9^2 + 9^2) = 0.9525$$

Для суммы случайных величин длительностей работ имеем:
$P\big(T \geq 1.1M(T)\big) = 0.5 - \Phi\left(\frac{e}{\vartheta T}\right)$, где Φ – это функция Лапласа (табулированный интеграл вероятности):

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_0^x e^{-\frac{t^2}{2}}dt$$

По условию время выполнения комплекса работ не должно превышать детерминированное значение на 10%, то есть на $e = 47 * 0.1 = 4.7$.

$$P\big(T \geq 1.1M(T)\big) = 0.5 - \Phi\left(\frac{e}{\vartheta T}\right) = 0.5 - \Phi\left(\frac{4.7}{\sqrt{0.9525}}\right) = 0.5 - \Phi(4.8) = 0.5 - 0.499999 \approx 0$$

Результат показывает, что шанс отклониться от математического ожидания времени выполнения более чем на 10% крайне мал.

2.8. Представить пошаговую процедуру имитационного моделирования расписания по схеме событий с учетом числа исполнителей и решающего правила ранжирования работ из числа возможных. По результатам моделирования построить диаграмму Гантта.

Правило выбора работ:
- Решающее правило: Короткие работы — вперед
- Число исполнителей: 3

Параметры:
- T – системное время.
- $\Omega_{\text{Р}}(T)$ – ранжированный список возможных работ.
- $N(T)$ – список выполняемых на момент времени Т работ: начатых, но не завершенных к этому моменту.
- $Z(T)$ – список времен освобождения ресурсов на момент времени Т.
- $B(T)$ – список выполненных на момент времени Т работ.
- $I(T)$ – список осуществленных событий.
- IJ – множество дуг-работ, исходящих из осуществленных событий.
- r_{sJOB} – список работ, выполняемых ресурсом s.
- r_{sSTART} – список моментов начала работ, выполняемых ресурсом s.
- $r_{sFINISH}$ – список моментов окончания работ, выполняемых ресурсом s.

T	$\Omega_{P}(T)$	$N(T)$	$Z(T)$	$B(T)$	$I(T)$	IJ	r_{sJOB}	r_{sSTART}	$r_{sFINISH}$
	Что доступно	Выполняется сейчас	Время на выполнение	Список выполненных	Какие узлы закрыли	Все осущ работы+ доступ	Кто и что делает		
0	12, 13, 14, 15	∅	∅	∅	1	12, 13, 14, 15	∅	∅	∅
0	14	12 13 15	9 9 5	∅	1	12, 13, 14, 15	1: 12 2: 13 3: 15	0	9 9 5
5	∅	12 13 14	9 9 9	15	1	12, 13, 14, 15	1: 12 2: 13 3: 14	0 0 5	9 9 14
9	25, 35	14	9	12, 13, 15	1, 2	12, 13, 14, 15, 25, 35	3: 14	5	14
9	∅	14 25 35	9 6 7	12, 13, 15	1, 2	12, 13, 14, 15, 25, 35	1: 25 2: 35 3: 14	9 9 5	15 16 14
14	46	25 35	6 7	12, 13, 14, 15	1, 2, 3	12, 13, 14, 15, 25, 35, 46	1: 25 2: 35	9 9	15 16
14	∅	25 35 46	6 7 5	12, 13, 14, 15	1, 2, 3, 4	12, 13, 14, 15, 25, 35, 46	1: 25 2: 35 3: 46	9 9 14	15 16 19
15	∅	35 46	7 5	12, 13, 14, 15, 25	1, 2, 3, 4	12, 13, 14, 15, 25, 35, 46	2: 35 3: 46	9 14	16 19
16	56, 57, 58	46	5	12, 13, 14, 15, 25, 35	1, 2, 3, 4	12, 13, 14, 15, 25, 35, 46, 56, 57, 58	3: 46	14	19
16	58	46 56 57	5 8 8	12, 13, 14, 15, 25, 35	1, 2, 3, 4	12, 13, 14, 15, 25, 35, 46, 56, 57, 58	1: 56 2: 57 3: 46	16 16 14	24 24 19
19	58	56 57	8 8	12, 13, 14, 15, 25, 35, 46	1, 2, 3, 4	12, 13, 14, 15, 25, 35, 46, 56, 57, 58	1: 56 2: 57	16 16	24 24
19	∅	56 57 58	8 8 9	12, 13, 14, 15, 25, 35, 46	1, 2, 3, 4, 5	12, 13, 14, 15, 25, 35, 46, 56, 57, 58	1: 56 2: 57 3: 58	16 16 19	24 24 28

24	67, 68, 69	58	9	12, 13, 14, 15, 25, 35, 46, 56, 57	1, 2, 3, 4, 5	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69	3: 58	19	28
24	69	58 67 68	9 5 8	12, 13, 14, 15, 25, 35, 46, 56, 57	1, 2, 3, 4, 5	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69	1: 67 2: 68 3: 58	24 24 19	29 32 28
28	69	67 68	5 8	12, 13, 14, 15, 25, 35, 46, 56, 57, 58	1, 2, 3, 4, 5, 6	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69	1: 67 2: 68	24 24	29 32
28	∅	67 68 69	5 8 9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58	1, 2, 3, 4, 5, 6	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69	1: 67 2: 68 3: 69	24 24 28	29 32 37
29	78, 79	68 69	8 9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67	1, 2, 3, 4, 5, 6, 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	2: 68 3: 69	24 28	32 37
29	78	67 68 79	9 8 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67	1, 2, 3, 4, 5, 6, 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	1: 79 2: 68 3: 69	29 24 28	36 32 37
32	78	69 79	9 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68	1, 2, 3, 4, 5, 6, 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	1: 79 3: 69	29 28	36 37
32	∅	69 79 78	9 7 9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68	1, 2, 3, 4, 5, 6, 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	1: 79 2: 78 3: 69	29 32 28	36 41 37
36	∅	69 78	9 9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 79	1, 2, 3, 4, 5, 6, 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	2: 78 3: 69	32 28	41 37
37	∅	78	9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 79	1, 2, 3, 4, 5, 6, 7	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	2: 78	32	41
41	89	∅	∅	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	1, 2, 3, 4, 5, 6, 7, 8	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79. 89	∅	∅	∅

41	∅	89	9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79	1, 2, 3, 4, 5, 6, 7, 8	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79. 89	1: 89	41	50
50	∅	∅	∅	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79, 89	1, 2, 3, 4, 5, 6, 7, 8, 9	12, 13, 14, 15, 25, 35, 46, 56, 57, 58, 67, 68, 69, 78, 79. 89	∅	∅	∅
I(50) = I – конец работы									

По результатам таблицы составим диаграмму Ганта:

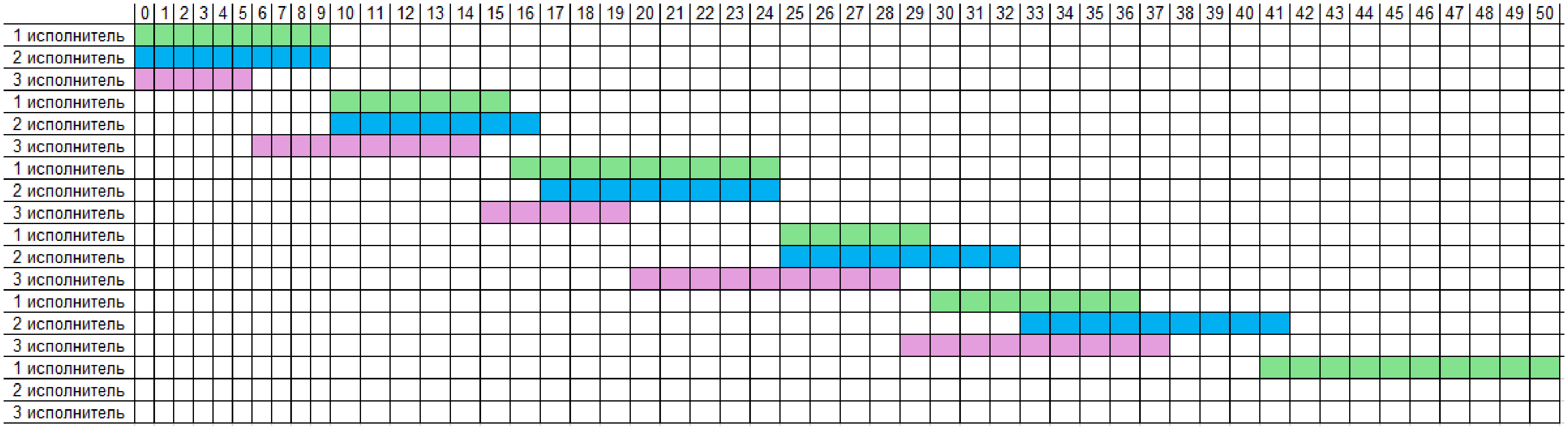

Рис. 2.6. Диограмма Ганта.

3. Вывод:

В ходе расчетной работы были получены навыки по построению модели расписания, а также применению методов линейного и динамического программирования в рамках поставленной задачи.

4. Приложение:

Листинг на github: https://github.com/IlyaFonichev/SADM_6_1