

Министерство образования и науки Республики Беларусь

Гомельский государственный университет
им.Ф.Скорины

кафедра математических проблем управления

"Исследование операций"
Лабораторный практикум

Гомель, 1998

Составители: Т.П.Бышик, В.Д.Левчук, В.Л.Мережа, Н.Б.Осипенко

Рецензенты: д.т.н., профессор В.В.Можаровский, к.ф.-м.н., доцент
В.Н.Семенчук

Рекомендовано советом математического факультета.

В лабораторный практикум включены работы: решение задачи о назначениях венгерским методом; решение задачи о коммивояжере методом ветвей и границ; решение целочисленных линейных задач методом отсечения Гомори; решение матричных игр; расчет и анализ параметров сетевого графика; исследование систем массового обслуживания с параллельными обслуживающими каналами и ограниченным количеством мест в очереди.

Каждая из работ содержит постановку задачи, алгоритм решения и иллюстрированный пример.

© Гомельский университет им. Ф. С Сорины, 1997

© Gomel University Press, 1997

Лабораторная работа №1

Решение задачи о назначениях венгерским методом

Постановка задачи: пусть имеется n видов работ и n исполнителей, каждый из которых может выполнить любую из этих работ; c_{ij} — эффективность выполнения i -ой работы j -ым исполнителем, $i, j = \overline{1, n}$. Из параметров c_{ij} можно составить $(n \times n)$ матрицу стоимостей (эффективностей) C . Найти такую расстановку исполнителей, чтобы суммарный эффект их труда был наибольшим.

Математическая модель задачи: введем переменные

$$x_{ij} = \begin{cases} 1, & \text{работа } i \text{ выполняется } j\text{-м исполнителем,} \\ 0, & \text{в противном случае.} \end{cases}$$

Получаем задачу:

$$L(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max \quad \sum_{j=1}^n x_{ij} = 1 \quad (1), \quad \sum_{i=1}^n x_{ij} = 1 \quad (2), \quad i, j = \overline{1, n}$$

Ограничения (1) означают, что на каждую работу должен быть назначен только один исполнитель. Ограничения (2) означают, что каждый исполнитель выполняет только одну работу.

Определение. Система нулевых элементов матрицы, обладающая тем свойством, что никакая пара из них не лежит в одной строке или в одном столбце, называется *системой независимых нулей (СНИ)*.

Идея метода.

Венгерский метод состоит в преобразовании исходной задачи с матрицей C в эквивалентную ей задачу минимизации с матрицей C'' . Если в матрице C'' имеется система из n независимых нулей, то решением задачи будет матрица X , у которой на местах, соответствующим независимым нулям, стоят 1, а остальные элементы являются нулевыми. Таким образом алгоритм направлен на построение СНИ путем эквивалентных, преобразований матрицы C'' .

Алгоритм метода.

0. Преобразуем матрицу C в матрицу C'' следующим образом:

a) $c'_{ij} = \max\{c_{ij}\} - c_{ij}, \quad i, j = \overline{1, n};$

b) $c'' = c'_{ij} - \min_{1 \leq j \leq n} \{c'_{ij}\}, \quad i, j = \overline{1, n}.$

Приходим к задаче: $L'(x) = \sum_{i=1}^n \sum_{j=1}^n c''_{ij} x_{ij} \rightarrow \min \quad \sum_{j=1}^n x_{ij} = 1, \quad \sum_{i=1}^n x_{ij} = 1, \quad i, j = \overline{1, n}$

1. Строим систему независимых нулей, помечая их $*$. Пусть число таких элементов равно k . Если $k=n$, то пишем ответ — матрицу X и $L(X)$. Если $k \neq n$, то помечаем столбцы, содержащие 0^* сверху знаком «+». Все элементы в этих столбцах назовем *выделенными элементами (ВЭ)* (аналогично, так назовем элементы и в строках со знаком «+»).

2. Среди невыделенных элементов ищутся нулевые. Если они есть, то помечаем один из них $0'$ и переходим на 3. Если нет, то — 5.

3. Если в строке, содержащей только что найденный (U нет 0^* , то — переход на 4. Иначе с соответствующего столбца снимается знак выделения и выделяется строка, содержащая $0'$ и осуществляется переход на 2.

4. Начиная с только что найденного $0'$ строится цепочка нулевых элементов матрицы по следующему правилу: исходный $0'$; 0^* , лежащий в одном столбце с этим $0'$; $0'$, лежащий в строке с только что пройденным 0^* и т.д. Заканчивается цепочка $0'$. После построения цепочки штрихи у нулевых элементов этой цепочки заменяются на $*$, старые $*$ в цепочке уничтожаются. Убираются знаки выделения строк и столбцов. $*$ не входящие в цепочку переписываются. Далее переход на 1.

5. Ищем минимальный ненулевой неВЭ и обозначаем h . Затем, прибавляем h к элементам, которые стоят на пересечении выделенных строк и выделенных столбцов; отнимаем h от элементов, которые стоят на пересечении невыделенных строк и не выделенных столбцов; остальные элементы переписываем без изменений. Возвращаемся к 2.

Пример. Решить задачу о назначениях с матрицей C венгерским методом.

$$C = \begin{pmatrix} 9 & 6 & 5 & 8 \\ 4 & 8 & 6 & 2 \\ 6 & 7 & 9 & 4 \\ 2 & 7 & 3 & 1 \end{pmatrix}$$

Решение.

$$C = \begin{pmatrix} 9 & 6 & 5 & 8 \\ 4 & 8 & 6 & 2 \\ 6 & 7 & 9 & 4 \\ 2 & 7 & 3 & 1 \end{pmatrix} \rightarrow C' = \begin{pmatrix} 0 & 2 & 4 & 0 \\ 5 & 0 & 3 & 6 \\ 3 & 1 & 0 & 4 \\ 7 & 1 & 6 & 7 \end{pmatrix} \rightarrow C'' = \begin{pmatrix} 0^* & 2 & 4 & 0' \\ 5 & 0^* & 3 & 6 \\ 3 & 1 & 0^* & 4 \\ 6 & 0 & 5 & 6 \end{pmatrix} +$$

1. В матрице C'' строим и подсчитываем число независимых нулей; $k = 3 < 4$, значит столбцы, содержащие 0^* помечаем +.

2. Среди невыделенных элементов ищется нулевой, это элемент c_{14} помечаем его '.

3. В первой строке есть 0^* , поэтому с первого столбца надо снять знак выделения \oplus и отметить знаком + первую строку.

4. Среди неВЭ нет нулевых.

5. Выбираем минимальный элемент среди неВЭ $c_{31} = h = 3$ и

— прибавляем h к элементам, которые стоят на пересечении ВЭ строк и ВЭ столбцов;

— вычитаем h от элементов, которые стоят на пересечении неВЭ строк и неВЭ столбцов;

— оставляем без изменения остальные элементы. Получаем матрицу:

$$C''' = \begin{pmatrix} 0^* & 5 & 7 & 0' \\ 2 & 0^* & 3 & 3 \\ 0' & 1 & 0^* & 1 \\ 3 & 0 & 5 & 3 \end{pmatrix} +$$

2. Среди неВЭ есть нулевой: $c_{31} = 0$, помечаем его '.

3. В третьей строке существует 0^* , значит снимаем с третьего столбца знак \oplus и помечаем третью строку +.

2. Среди неВЭ нет нулей.

5. Выбираем в C''' $c_{21} = h = 2$ и снова строим новые нули, получаем матрицу:

$$C^{IV} = \begin{pmatrix} 0^* & 7 & 7 & 0' \\ 0' & 0^* & 1 & 1 \\ 0' & 3 & 0^* & 1 \\ 1 & 0' & 3 & 1 \end{pmatrix} +$$

2. Ищем 0 среди НЭ: $c_{21} = 0$; помечаем его 0'.

3. Во 2 строке есть 0*, снимаем знак \oplus со 2 столбца и помечаем знаком + 2 строку.

2. В 4 строке есть $c_{42} = 0$, помечаем его 0'.

3. В 4 строке нет 0*.

4. Строим цепочку, начиная с $c_{42} : \{c_{42}, c_{22}, c_{21}, c_{11}, c_{14}\}$ В цепочке заменяем на *, а старые * убираем. Получаем C^{IV}

$$C^V = \begin{pmatrix} 0 & 7 & 7 & 0^* \\ 0^* & 0 & 1 & 1 \\ 0 & 3 & 0^* & 1 \\ 1 & 0^* & 3 & 1 \end{pmatrix}$$

1. Подсчитываем число $k = 4 = n$, значит записываем ответ:

$$X = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad L(X) = 8 + 4 + 9 + 7 = 28(\text{ед.})$$

Задание: Решить задачу о назначениях венгерским методом.

Лабораторная работа №2

Решение задачи о коммивояжере методом ветвей и границ

Постановка задачи. Бродячий торговец должен посетить n городов и вернуться в исходный. Маршрут должен проходить через каждый город, причем один и только один раз. Расстояния (транспортные издержки, время на переезд и т.д.) между городами известны — c_{ij} $i = \overline{1, n}; j = \overline{1, n}$. Требуется отыскать самый короткий (либо самый дешевый, либо самый быстрый и т.д.) маршрут.

Математическая модель:

Пусть $x_{ij} = \begin{cases} 1, & \text{в маршруте есть переезд из } i \text{ городов в } j; \\ 0, & \text{в противном случае.} \end{cases}$

$$L(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (1)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad i = \overline{1, n} \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad j = \overline{1, n} \quad (3)$$

$x = (x_{ij}, i = \overline{1, n}, j = \overline{1, n})$ — некоторый маршрут, $L(x)$ — суммарные издержки; (2) означает, что из каждого города торговец выезжает только раз; (3) означает, что в каждый город он въезжает один раз. К математической модели надо добавить дополнительное ограничение, исключающее замкнутые подциклы (подмаршруты) в маршрутах

Реализация метода ветвей и границ.

Рассмотрим дерево всевозможных маршрутов (см, рис.1, $n = 5$). Так как маршрут замкнут, то все равно из какого города начинать. Каждому маршруту соответствует ветвь дерева, начинающаяся на 1 уровне дерева в городе 1 и заканчивающаяся на последнем уровне в одном из городов. Ясно, что всего маршрутов $(n - 1)!$.

- С помощью дерева любое подмножество маршрутов можно разбить на более мелкие подмножества. Например, (см.рис.1), все множество маршрутов можно разбить на четыре подмножества с начальными участками: $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$ и $\{1, 5\}$. Причем каждое такое подмножество включает $(5-2)! = 3! = 6$ маршрутов дальнейших продолжений начального участка.

- Для каждого подмножества маршрутов может быть подсчитана эффективная нижняя граница по следующему способу.

Например, пусть задана матрица транспортных издержек

$$C = \begin{pmatrix} - & 8 & 18 & 14 & 8 \\ 17 & - & 21 & 7 & 11 \\ 20 & 9 & - & 8 & 12 \\ 5 & 31 & 12 & - & 9 \\ 4 & 3 & 15 & 6 & - \end{pmatrix}$$

Для нахождения оценки начального участка $\{1, 2\}$ — $\xi\{1, 2\}$ строим подматрицу матрицы C вычеркивая в ней первую строку и второй столбец. У полученной подматрицы \bar{C} находим минимальные элементы у строк. Отнимаем

их от элементов соответствующих строк и получаем подматрицу \bar{C} . Находим у нее минимальные элементы у столбцов. В качестве оценки ξ берется сумма минимальных элементов строк подматрицы \bar{C} и столбцов \bar{C} и длины начального участка

$$\bar{C} = \begin{pmatrix} 17 & 21 & 7 & 11 \\ 20 & - & 8 & 12 \\ 5 & 12 & - & 9 \\ 4 & 15 & 6 & - \end{pmatrix} \begin{matrix} 7 \\ 8 \\ 5 \\ 4 \end{matrix} \quad \bar{C} = \begin{pmatrix} 10 & 14 & 0 & 4 \\ 12 & - & 0 & 4 \\ 0 & 7 & - & 4 \\ 0 & 9 & 2 & - \\ 0 & 7 & 0 & 4 \end{pmatrix}; \quad c_{12} = 8$$

$$\xi\{1,2\} = (7+8+5+4+0+7+0+4) + 8 = 43$$

Для нахождения оценки для участка $\{1,5,3\}$ у матрицы C вычеркиваются строки 1 и 5 и столбцы 5 и 3.

В общем случае для подсчета оценки произвольного участка у матрицы C вычеркивают строки для городов из которых торговец выезжает и столбцы для городов куда он приезжает.

- Для вычисления рекорда итерации выбираются один либо несколько маршрутов. Вычисляются их длины. Рекордом r будет длина самого короткого из них. Соответствующий маршрут будет рекордным. На итерациях рекорд может быть улучшен.

- Если для некоторого подмножества маршрутов $\xi \geq r$, то его (и соответствующую ветвь дерева) можно исключить как заведомо не содержащую лучшего маршрута, чем рекордный. Решение задачи заканчивается когда будут отсечены все ветви.

Пример (для матрицы (4)).

0 итерация. $S_0 = \{X\}$.

Вычислим $\xi(X) = 8 + 7 + 8 + 5 + 3 + 7 = 38$. Вычислим длину одного из маршрутов $L(1,5,2,4,3,1) = 8 + 3 + 7 + 12 + 20 = 50$. Он и будет пока рекордным. Так как $\xi_0 = 38 < 50 = r_0$, то разбиваем X на 4 подмножества с начальными участками: $(1,2), (1,3), (1,4), (1,5)$.

Вычислим их оценки: $\xi(1,2) = 43; \xi(1,3) = 45; \xi(1,4) = 59; \xi(1,5) = 38;$

Строим список множеств для 1 итерации: $S_1 = \{(1,2), (1,3), (1,5)\}$. Множество $(1,4)$ отсекается, так как $\xi(1,4) = 59 > 50 = r_0$.

1 итерация. Вычислим $\xi_1 = \min\{\xi(1,2), \xi(1,3), \xi(1,5)\} = 38 \quad r_1 = r_0 = 50$.

Так как $r_1 = 50 > 38 = \xi_1$ то продолжим итерации. Разбиваем множество с наименьшей оценкой $(1,5)$ на 3 подмножества: $(1,5,2), (1,5,3), (1,5,4)$. Вычислим их оценки: $\xi(1,5,2) = 38; \xi(1,5,3) = 44; \xi(1,5,4) = 49$. Имеем

$S_2 = \{(1,2)(1,3), (1,5,2)(1,5,3), (1,5,4)\}$

2 итерация. $\xi_2 = 38, r_2 = r_0 = 50$ Разбиваем $(1,5,2)$: $(1,5,2,3), (1,5,2,4)$. У этих множеств лишь по одному элементу не хватает до полного маршрута. Поэтому вычислим длину маршрутов $L(1,5,2,3,4,1) = 45, L(1,5,2,4,3,1) = 50$. Получаем новый рекорд 45 и имеем $S_3 = \{(1,2)(1,5,3)\}$. Остальные подмножества отсекаются.

3 итерация. $\xi_3 = 43, r_3 = 45$. Разбиваем $(1,2)$: $(1,2,3), (1,2,4), (1,2,5)$.
 $\xi(1,2,3) = 48, \xi(1,2,4) = 43, \xi(1,2,5) = 43$ Имеем $S_4 = \{(1,2,4), (1,5,3), (1,2,5)\}$.

4 итерация. $\xi_4 = 43, r_4 = 45$. Разбиваем $(1,2,4)$: $(1,2,4,3), (1,2,4,5)$. Вычислим
 $L(1,2,4,3,5,1) = 43, L(1,2,4,5,3,1) = 59$. Новый рекорд 43.

5 итерация. Так как $\xi_5 = r_5 = 43$, то маршрут $(1,2,4,3,5,1)$ — оптимальный, $L^0 = 43$.

Дерево вариантов этой задачи имеет вид (см. рис. 2):

Задание. Решить задачу о коммивояжере методом ветвей и границ. Построить дерево вариантов для своей задачи.

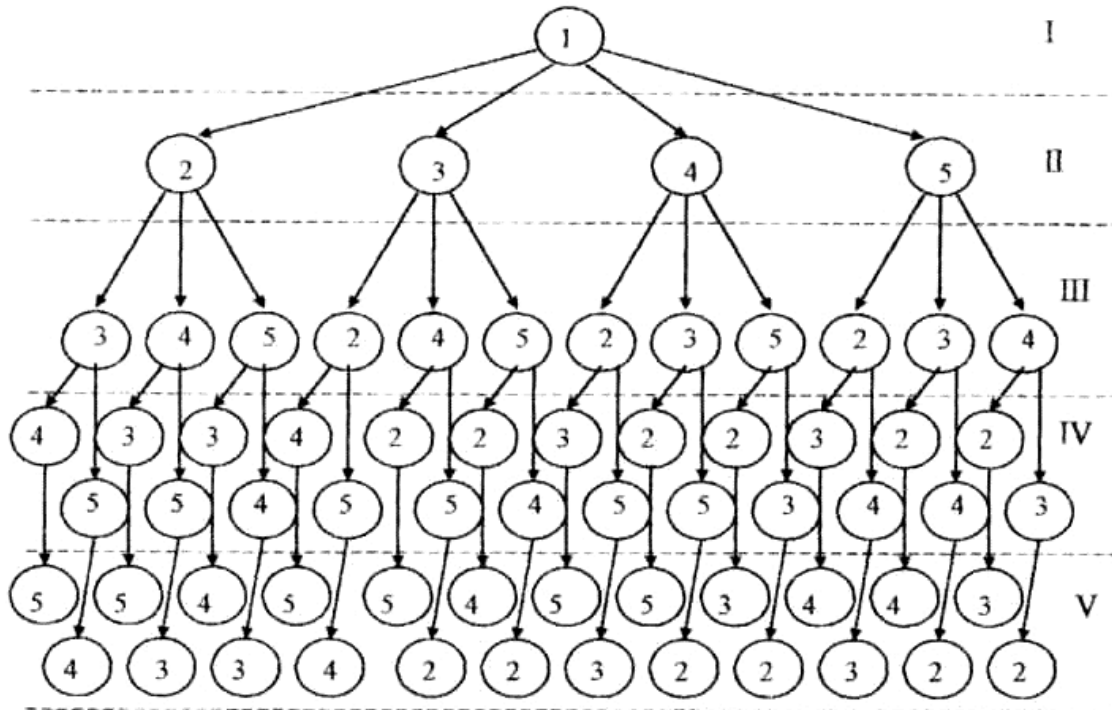


Рис.1

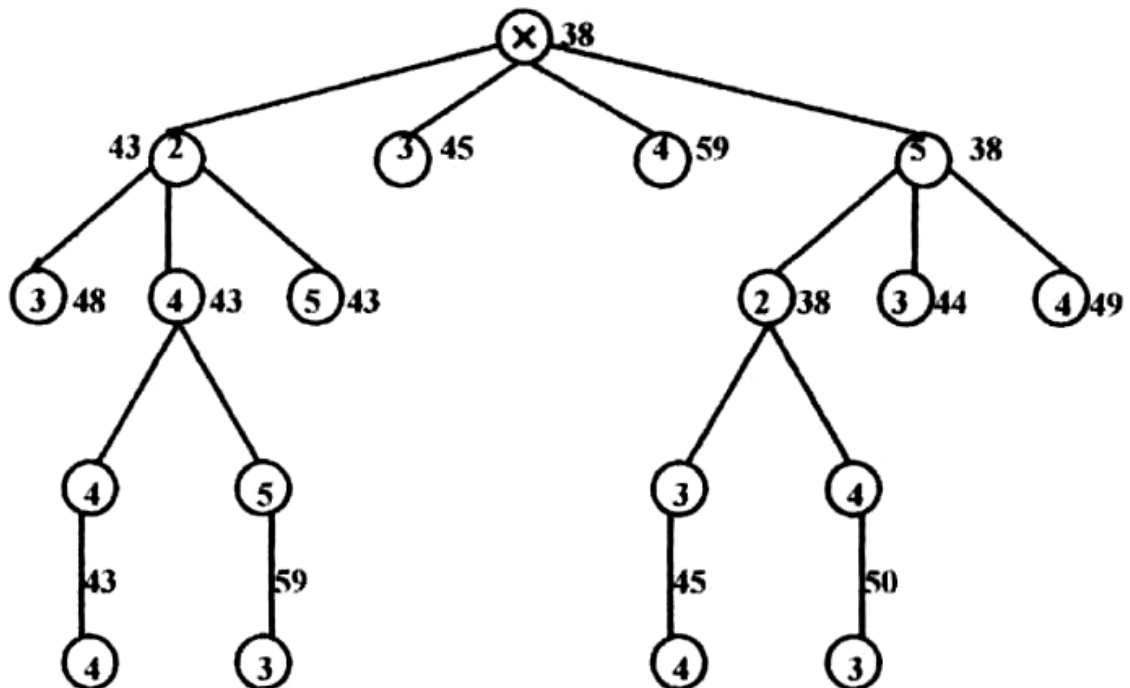


Рис.2.Дерево вариантов метода.

Лабораторная работа №3

Решение целочисленных линейных задач методом отсечения Гомори

Постановка задачи: найти $x \in R^n$ такое, что $z = c'x \rightarrow \max$ при ограничениях $Ax = b, x \geq 0, x_i \text{ — целые числа, } i = \overline{1, n}, b \in R^m, c \in R^n, A - (m \times n)\text{-матрица.}$

Идея метода. Сначала решается соответствующая непрерывная задача (условия целочисленности не учитываются) симплекс-методом. Если полученный оптимальный план целочислен, то он будет решением исходной задачи. Если нет, то по одной из дробных компонент этого плана строится дополнительное ограничение (сечение Гомори), которое "отсекает" его от множества планов, а все целочисленные остаются, непрерывная задача с дополнительным ограничением снова решается. При этом используется результат решения предыдущей задачи (последняя симплекс таблица) и ее удобно решать двойственным симплекс-методом. Если новый оптимальный план целочислен, то получено решение исходной задачи, в противном случае операция (построения сечения Гомори) повторяется. И т.д. Доказано, что метод Гомори через конечное число таких операций позволяет построить решение задачи.

Пример. Пусть мы имеем задачу целочисленного линейного программирования.

$$z = x_1 + x_2 \rightarrow \max$$

$$2x_1 + x_2 + x_3 = 6$$

$$2x_1 + 3x_2 + x_4 = 9, x_i \geq 0 \text{ — целые числа } i = \overline{1, n}$$

Применим к непрерывной задаче симплекс-метод. Получим оптимальный план: $x^0 = \left(\frac{9}{4}, \frac{3}{2}, 0, 0\right) \quad z_{\max} = 3\frac{3}{4}$.

Обозначим:

$[\alpha]$ — целая часть числа α ;

$\{\alpha\} = \alpha - [\alpha]$ — дробная часть α .

Последняя симплекс-таблица имеет вид:

a _B	b	a ₁	a ₂	a ₃	a ₄
a ₁	9/4	1	0	3/4	-1/4
a ₂	3/2	0	1	-1/2	1/2
Δ	15/4	0	0	1/4	1/4

Так как план X^0 не целочислен применим к задаче метод Гомори.

1. Из b_2 выбираем число с наибольшей дробной частью. Так как $\max \left[\left\{ \frac{9}{4} \right\} = \frac{1}{4}; \left\{ \frac{3}{2} \right\} = \frac{1}{2} \right] = \frac{1}{2}$, то по II строке строим отсечение Гомори по формуле:

$$\{b_i\} - \sum_{j \in I_H} \{a_{ij}\} x_{ij} \leq 0$$

$$\text{имеем } \left\{ \frac{3}{2} \right\} - \left(\left\{ -\frac{1}{2} \right\} x_3 + \left\{ \frac{1}{2} \right\} x_4 \right) \leq 0 \quad \frac{1}{2} - \left(\frac{1}{2} x_3 + \frac{1}{2} x_4 \right) \leq 0; \quad -\frac{1}{2} x_3 - \frac{1}{2} x_4 \leq -\frac{1}{2}$$

Приводим это ограничение к равенству введя дополнительную переменную $x_5 \geq 0$.

$$-\frac{1}{2}x_3 - \frac{1}{2}x_4 + x_5 = -\frac{1}{2}$$

Так как $b_5 = -\frac{1}{2}$, значит строим двойственную симплекс-таблицу и решаем полученную задачу двойственным симплекс-методом ($\delta = \Delta$).

a_B	b	a_1	a_2	a_3	a_4	a_5
a_1	9/4	1	0	3/4	-1/4	0
a_2	3/2	0	1	-1/2	1/2	0
a_5	-1/2	0	0	-1/2	-1/2	1
Δ	15/4	0	0	1/4	1/4	0
σ				1/2	1/2	
a_1	3/2	1	0	0	-1	3/2
a_2	2	0	1	0	1	-1
a_3	1	0	0	1	1	-2
Δ	7/2	0	0	0	0	1/2

3. Строим сечение Гомори для I строки:

$$\left\{ \frac{3}{2} \right\} - \left(\{-1\}x_4 + \left\{ \frac{3}{2} \right\}x_5 \right) \leq 0 \quad \frac{1}{2} - \left(0^*x_4 + \frac{1}{2}x_5 \right) \leq 0, \quad \frac{1}{2} - \frac{1}{2}x_5 \leq 0; \quad -\frac{1}{2} + x_5 + x_6 = -\frac{1}{2}$$

4. Строим новую двойственную симплекс-таблицу.

a_1	3/2	1	0	0	-1	3/2	0
a_2	2	0	1	0	1	-1	0
a_3	1	0	0	1	1	-2	0
a_6	-1/2	0	0	0	0	-1/2	1
Δ	7/2	0	0	0	0	1/2	0
a_1	0						
a_2	3						
a_3	3						
a_5	1						
Δ	3	0	0	0	0	0	1

$$x^* = (0, 3, 3, 0), \quad z_{\max} = 3$$

Задание. Решить задачу методом отсечения Гомори.

Лабораторная работа №4

Решение матричных игр

Рассмотрим конечную игру двух лиц с нулевой суммой. Пусть игрок 1 имеет m ходов — A_1, A_2, \dots, A_m , а игрок 2 (противник) — n ходов B_1, B_2, \dots, B_n . Если игрок 1 выбрал ход A_i , игрок 2 выбрал B_j , то они получают выигрыш, соответственно: $u_1(A_i, B_j), u_2(A_i, B_j)$ причем $u_1(A_i, B_j) = -u_2(A_i, B_j) = a_{ij}$. Из элементов a_{ij} можно составить матрицу $m \times n$

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

которая называется платежной. Тогда ходу A_i игрока 1 будет соответствовать выбор им i -ой строки матрицы A , а ходу B_j для игрока 2 — j -ой столбца. Цель игрока 1 максимизировать u_1 игрока 2 — максимизировать u_2 (или минимизировать $-u_2$). Игроки выбирают свои ходы не зная какие ходы выберет противник. Считая противника разумным, игроки могут придерживаться осторожных (перестраховочных) стратегий. Игрок 1 в каждой строке находит минимальный элемент $\alpha_i = \min_j a_{ij}, i = \overline{1, m}$, а затем находит свой гарантированный выигрыш $\alpha = \max_i \alpha_i$. Выбор соответствующий строки (строк) будет его максимальной стратегией (при любом поведении игрока 2 он получит не менее α). Аналогично игрок 2 находит $\beta_j = \max_i a_{ij}$ и $\beta = \min_j \beta_j$. Его мини-максной стратегией будет выбор соответствующего столбца (при любом поведении игрока a он обеспечит себе проигрыш не более β).

Если $\alpha = \beta$, то это значит, что существует элемент $a_{i^*j^*}$ который будет минимальным в строке i^* и максимальным в столбце j^* . Тогда стратегии A_{i^*} и B_{j^*} будут оптимальными для игроков 1 и 2. Величина $\gamma = \alpha = \beta$ называется *ценой игры*, а пара ходов (A_{i^*}, B_{j^*}) — *седловой точкой игры*.

Если седловой точки у игры нет и розыгрыши совершаются неоднократно, то естественно для игрока 1 попытаться увеличить α , а для игрока 2 — уменьшить β .

Смешанными стратегиями игроков 1 и 2 будем называть соответственно $p = (p_1, p_2, \dots, p_m)$ и $q = (q_1, q_2, \dots, q_n)$, где $p_i \geq 0, q_j \geq 0$ — вероятности (частоты) применения чистых стратегий A_i и B_j при этом $\sum_{i=1}^m p_i = 1, \sum_{j=1}^n q_j = 1$. Величина

$H(p, q) = \sum_{i=1}^m \sum_{j=1}^n p_i q_j a_{ij}$ — средний выигрыш (математическое ожидание игрока 1).

Основная теорема. Любая матричная игра имеет пару оптимальных смешанных стратегий $p^* = (p_i^* i = \overline{1, m}), q^* = (q_j^* j = \overline{1, n})$, обладающую свойством:

$$\sum_{i=1}^m \sum_{j=1}^n p_i q_j \alpha_{ij} \leq \sum_{i=1}^m \sum_{j=1}^n p_i^* q_j^* \alpha_{ij} \leq \sum_{i=1}^m \sum_{j=1}^n p_i^* q_j \alpha_{ij}$$

и цену игры $\gamma = H(p^*, q^*)$. (То есть если игрок 1 не придерживается p^* , а игрок 2 придерживается q^* тогда он может получить меньше γ . Соответственно для игрока 2.)

Будем считать, что все элементы платежной матрицы неотрицательны (если это не так, то можно ко всем ее элементам прибавить некоторое положительное число L , переводящее платежи в область неотрицательных значений; при этом решение игры p^* и q^* не изменится, цена игры лишь увеличится на L . Тогда $\gamma > 0$)

$$\text{Введем обозначения: } x_i = \frac{p_i}{\gamma}, i = \overline{1, m}; \quad y_j = \frac{q_j}{\gamma}, j = \overline{1, n};$$

Известно, что задача нахождения p^* и q^* сводится к паре взаимодвойственных задач:

$$\sum_{i=1}^m x_i \rightarrow \min, \quad \sum_{i=1}^m \alpha_{ij} x_i \geq 1, \quad x_i \geq 0; i = \overline{1, m} \quad (1)$$

$$\sum_{j=1}^n y_j \rightarrow \max, \quad \sum_{j=1}^n \alpha_{ij} y_j \leq 1, \quad y_j \geq 0; j = \overline{1, n} \quad (2)$$

Предположим, что мы нашли их решения (оно всегда существует)

$$x_0 = (x_i^0, i = \overline{1, m}) \quad y_0 = (y_j^0, j = \overline{1, n})$$

Тогда:

$$\frac{1}{\gamma} = \sum_{i=1}^m x_i^0 = \sum_{j=1}^n y_j^0; \quad p_i^* = x_i^0 \gamma, i = \overline{1, m}; \quad q_j^* = \gamma y_j^0, j = \overline{1, n}$$

Замечание. Задачу (2) удобно решать, применяя симплекс-метод, а задачу (1) — двойственный симплекс-метод. Однако на практике достаточно решить лишь одну из них, например задачу (2). Решение двойственной ей задачи (1) легко получить из последней симплекс-таблицы (см. пример).

Оптимальные стратегии p^* и q^* можно искать, применяя приближенный метод Брауна-Робинсон, который заключается в следующем. Розыгрыш осуществляется искусственно. Игроки попеременно выбирают свои стратегии (строки и столбцы матрицы A), в ответ на поведение противника. Этот выбор они делают по результатам всех предыдущих ходов. Суммируются выбранные ранее строки игроком 1 и столбцы игроком 2. После чего игрок 2 находит минимальное число в полученной строке и отвечает соответствующей ему стратегией, а игрок 1 находит максимальное в полученном столбце и делает соответствующий ему ход. Начинает розыгрыш, например, игрок 1 стратегией A_i . Данные сводятся в специальную таблицу (смотри пример). Там же записываются средний выигрыш a игрока 1 и средний проигрыш β игрока 2. После достаточного количества итераций (N), подсчитывают частоты $\bar{p}_i = \frac{S_i}{N}$, $\bar{q}_j = \frac{K_j}{N}$, $i = \overline{1, m}$, $j = \overline{1, n}$, S_i — количество стратегий A_i в них; K_j — количество стратегий B_j в них. Они и служат приближенными значениями для p^* и q^* . Средние арифметические a и β , служат оценками нижней и верхней

	Δ	3/7	5/7	0	1/7	0	1/7	2/7	0	
--	----------	-----	-----	---	-----	---	-----	-----	---	--

Оптимальное решение задачи (2): $y^* = \left(0; \frac{2}{7}; 0; \frac{1}{7}; 0; 0; \frac{3}{7}\right), T_{\max} = \frac{3}{7}$. Получим цену

игры $v = \frac{1}{T_{\max}} = \frac{7}{3}$ и компоненты оптимальной смешанной стратегии q_j^* игрока B :

$$q_1^* = v y_1^* = \frac{7}{3} * 0 = 0,$$

$$q_2^* = v y_2^* = \frac{7}{3} * \frac{2}{7} = \frac{2}{3},$$

$$q_3^* = v y_3^* = \frac{7}{3} * 0 = 0,$$

$$q_4^* = v y_4^* = \frac{7}{3} * \frac{1}{7} = \frac{1}{3}.$$

Так как базисным переменным y_5, y_6, y_7 задачи (2) соответствуют свободные переменные x_1, x_2, x_3 задачи (1), можем из Δ -строки последней симплекс-таблицы выписать оптимальное решение задачи (1):

$$x_1^* = \frac{1}{7}; x_2^* = \frac{2}{7}; x_3^* = 0; Z_{\min} = \frac{3}{7}$$

Теперь вычислим компоненты оптимальной смешанной стратегии p_i^* игрока A :

$$p_1^* = v x_1^* = \frac{7}{3} * \frac{1}{7} = 0,$$

$$p_2^* = v x_2^* = \frac{7}{3} * \frac{2}{7} = \frac{2}{3},$$

$$p_3^* = v x_3^* = \frac{7}{3} * 0 = 0.$$

Итак $p^* = \left(\frac{1}{3}; \frac{2}{3}; 0\right), q^* = \left(0; \frac{2}{3}; 0; \frac{1}{3}\right), v = 2\frac{1}{2}$.

3. Решим игру методом Брауна-Робинсона. Чтобы получить неплохое приближенное решение игры этим методом вам достаточно выполнить 20 итераций.

Вычисления удобно проводить в следующей таблице (см. таблицу 1)

Первый свой ход игрок A выбирает случайным образом, например, стратегию A_1 . В столбцы B_1-B_4 при первой итерации записывается строка исходной матрицы, соответствующая выбранной стратегии игрока 1, а затем суммируются строки, согласно выбранным в последствии стратегиям этот игрока. В столбцах A_1-A_3 заносятся суммы столбцов матрицы A соответственно выбранным ходам игрока 2. Стратегии (столбцы и строки) игроки 2 и 1 выбирают попеременно, в ответ на ход противника и так, чтобы минимизировать свой проигрыш (выбирается наименьшее число среди A_1-A_n) и максимизировать свой выигрыш (выбирается среди B_1-B_3), соответственно.

Каждый раз рассчитываются :

α_k — наименьший из накопленных выигрышей игрока 1 за k партий, деленный на число партий k ;

Лабораторная работа №5

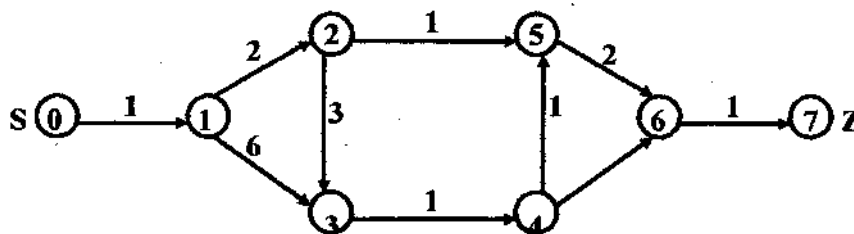
Расчет и анализ параметров сетевого графика

Постановка задачи. Пусть имеется проект некоторого комплекса работ. Проект задается перечнем работ, их продолжительностью и последовательностью их выполнения.

Представление проекта в виде сети называется *сетевым графиком*. Ориентированные дуги представляют работы. Вершины (события) представляют последовательность выполнения работ. В сетевом графике должно быть одно начальное и одно конечное событие.

Рассмотрим некоторые алгоритмы расчета ряда плановых показателей по сетевому графику работ, одновременно иллюстрируя их на примере.

Пример.



Данный график задает 10 работ: $(0,1), (1,2), (1,3), \dots$ и 8 событий, связанных с началом и завершением работ. Числа у дуг означают продолжительность перехода от одного события к другому.

1. Расчет наиболее раннего возможного времени наступления каждого события

$$t_p(j), \quad j = \overline{0,7}, \quad \langle p \rangle \text{ — раннее}$$

Формула:

$$t_p(0) = 0, \quad t_p(j) = \max_j [t_p(i) + t_{ij}]$$

где \max берется по событиям, для которых существует работа (i,j) .

Для нашего графика:

$$t_p(0) = 0, \quad t_p(1) = [t_p(0) + t_{01}] = 0 + 1 = 1,$$

$$t_p(2) = [t_p(1) + t_{12}] = 1 + 2 = 3,$$

$$t_p(3) = \max [t_p(1) + t_{13}, t_p(2) + t_{23}] = \max [1 + 6, 3 + 3] = 7,$$

$$t_p(4) = [t_p(3) + t_{34}] = 7 + 1 = 8, \quad (*)$$

$$t_p(5) = \max [t_p(2) + t_{25}, t_p(4) + t_{45}] = \max [3 + 1, 8 + 1] = 9,$$

$$t_p(6) = \max [t_p(4) + t_{46}, t_p(5) + t_{56}] = \max [8 + 1, 9 + 2] = 11$$

$$t_p(7) = [t_p(6) + t_{67}] = 11 + 1 = 12.$$

Таким образом установлено время начала каждой работы при соблюдении плановых сроков и время выполнения всех работ — 12 единиц времени (например, месяцев).

2. Определение критического пути.

Последовательность работ в сети $(i_1, i_2), (i_2, i_3), \dots$, в которой конечное событие каждой работы совпадает с начальным событием следующей работы, называется *путем* и обозначается перечислением событий $L = (i_1, i_2, \dots, i_k)$

Путь для которого начало совпадает с исходным событием, а конец с завершающим, называется *полным путем*. Указанный путь имеет 5 полных путей: $L_1 = (0,1,2,5,6,7)$, $L_2 = (0,1,2,3,4,6,7)$, $L_3 = (0,1,2,3,4,5,6,7)$, $L_4 = (0,1,3,4,6,7)$, $L_5 = (0,1,3,4,5,6,7)$. *Длиной пути* называется сумма продолжительностей составляющих его работ. Путь, имеющий максимальную длину, называется *критическим путем*. Поиск критического пути ведется от последнего события сети. В качестве события, предшествующего событию с номером j в критическом пути принимается событие с номером k , для которого

$$t_p(k) + t_{kj} = \max_i [t_p(i) + t_{ij}]$$

Эти номера k выбираем, начиная с последней строки формул (*): 7,6,5,4,3,1,0.

Таким образом, в нашем примере критический путь $L_{kp} = L_5 = (0,1,3,4,5,6,7)$. Длина $L_{kp} = 12$ (единицам времени).

3. Определение наиболее позднего возможного времени. Критический путь определяет общую продолжительность работ по созданию объекта, для которого составлен сетевой график. Чтобы сократить срок создания объекта нужно сократить время выполнения работ на критическом пути.

Возникает вопрос о резервах времени выполнения работ. Наличие таких резервов позволяет перераспределить силы для выполнения параллельных работ.

Для определения резервов времени предварительно вычисляется наиболее позднее возможное время начала выполнения каждого события, не изменяющее сроки выполнения всей разработки. Эти величины обозначим $t_n(j)$ ("n" — позднее). Для их определения предварительно рассчитываются величины $t'_n(j)$ — длина наибольшего пути от события j до конца разработки. Величины $t_n(j)$ определяются от последнего события по алгоритму:

$$t'_n(z) = 0, t'_n(j) = \max_i [t'_n(i) + t_{ij}]$$

Для данного примера имеем:

$$t'_n(7) = 0; \quad t'_n(6) = 1; \quad t'_n(5) = 3;$$

$$t'_n(4) = \max [t'_n(6) + t_{46}; t'_n(5) + t_{45}] = \max [3, 4] = 4;$$

$$t'_n(3) = 5; \quad t'_n(2) = \max [t'_n(5) + t_{25}; t'_n(3) + t_{23}] = \max [4, 8] = 8;$$

$$t'_n(1) = \max [t'_n(2) + t_{12}; t'_n(3) + t_{13}] = \max [10, 11] = 11; \quad t'_n(0) = 12.$$

Величины $t_n(j)$ определяются из соотношения:

$$t_n(o) = t_{kp} - t'_n(j)$$

$$\text{Имеем: } t_n(0) = 12 - 12 = 0; \quad t_n(1) = 1; \quad t_n(2) = 4; \quad t_n(3) = 7; \quad t_n(4) = 8; \quad t_n(5) = 9; \\ t_n(6) = 11; \quad t_n(7) = 12.$$

4. Определение полного резерва времени.

Для работы (i, j) полный резерв времени $R_n(i, j)$ определяется зависимостью:

$$R_n(i, j) = t_n(j) - t_p(i) - t_{ij}$$

и определяют насколько может быть затянуто выполнение работы (i, j) без удлинения сроков выполнения проекта.

Имеем:

$$R_n(0,1) = 0; R_n(1,2) = t_n(2) - t_p(1) - t_{12} = 4 - 1 - 2 = 1;$$

$$R_n(1,3) = 0; R_n(2,3) = 1; R_n(2,5) = 5; R_n(3,4) = 0;$$

$$R_n(4,5) = 0; R_n(4,6) = 1; R_n(5,6) = 0; R_n(6,7) = 0.$$

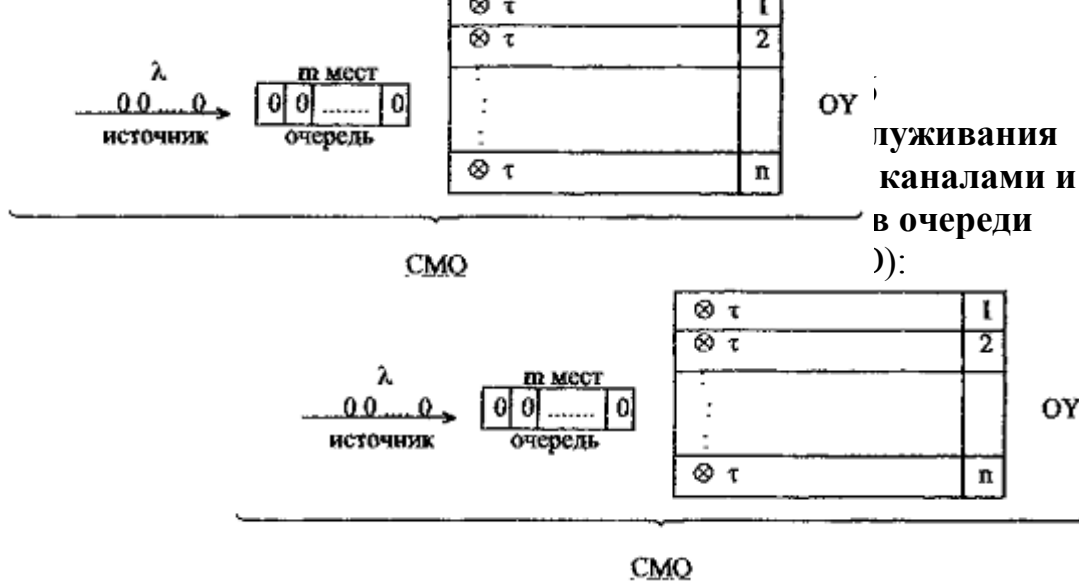
Для работ критического пути $R_n(i, j) = 0$. Использование одного из полных резервов работ может влиять на величину другого.

Так $R_n(1,2) = 1, R_n(2,5) = 5$. Но если использовать резерв $R_n(1,2)$, то величина $R_n(2,5)$ станет равной 4.

5. Свободный резерв времени. $P_c(i, j) = t_p(j) - t_n(i) - t_{ij}$

Если для работы (i, j) выполняется условие: $t_n(i) + t_{ij} < t_p(i)$, то работа обладает свободным резервом времени $P_c(2,5) = 4; P_c(4,6) = 1$. Использование величины $p_c(i, j)$ не влияет на величину резервов времени остальных работ сети.

Задание. Произвести расчет показателей сетевого графика. Сделать выводы.



На входе поток требований с интенсивностью λ , в очереди m мест, в ОУ n каналов, обслуживающих с интенсивностью $\mu = \frac{1}{\tau}$, где τ — среднее время обслуживания одной заявки. Если все $n+m$ мест в СМО заняты, то требование покидает ее.

Обозначим через $E_k, k = \overline{0, n+m}$ состояние, при котором в системе находится k требований, а через $p_k(t)$, соответствующую вероятность. Система уравнений для определения вероятностей состояний:

$$(1) \begin{cases} p'_0(t) = -\lambda p_0(t) + \mu p_1(t), \\ p'_k(t) = \lambda p_{k-1}(t) - (\lambda + k\mu) p_k(t) + (k+1)\mu p_{k+1}(t), k = \overline{1, n-1} \\ p'_k(t) = \lambda p_{k-1}(t) - (\lambda + n\mu) p_k(t) + n\mu p_{k+1}(t), k = \overline{n, n+m-1} \\ p'_{n+m}(t) = \lambda p_{n+m-1}(t) - n\mu p_{n+m}(t) \end{cases}$$

Решая эту систему дифференциальных уравнений возможно найти $p_k(t)$ для каждого $t > 0$ при некоторых начальных условиях $p_i(0), i = \overline{0, n+m}$. Так как в системе существует предельные установившиеся состояния $p_k(t) \rightarrow p_k$ при $t \rightarrow \infty$, то для них в пределе получаем систему ($p'_k(t) \rightarrow 0$ при $t \rightarrow \infty$).

$$(2) \begin{cases} -\lambda p_0 + \mu p_1 = 0 \\ \lambda p_{k-1} - (\lambda + k\mu) p_k + \mu(k+1) p_{k+1} = 0, k = \overline{1, n-1} \\ \lambda p_{k-1} - (\lambda + n\mu) p_k + n\mu p_{k+1} = 0, k = \overline{n, n+m-1} \\ \lambda p_{n+m-1} - n\mu p_{n+m} = 0. \end{cases}$$

с нормирующим условием $\sum_{k=0}^{n+m} p_k = 1$.

Решая эту систему, получим

$$\begin{cases} p_k = \frac{\rho^k}{\lambda_k!} p_0, k = \overline{1, n-1} & p_k = \frac{\rho^k}{n^{k-m} n!} p_0, k = \overline{n, n+m} \\ p_0 = \left[\sum_{k=0}^n \frac{\rho^k}{k!} + \frac{\rho^n \left[\left(\frac{\rho}{n} \right)^{m+1} - \frac{\rho}{n} \right]}{n! \left(\frac{\rho}{n} - 1 \right)} \right]^{-1}, \text{ где } \rho = \frac{\lambda}{\mu}. \end{cases}$$

Отсюда возможно определить основные характеристики системы:

1) p_0 — вероятность того, что все обслуживающие каналы свободны;

2) $p_k, k = \overline{1, n}$ — вероятность того, что обслуживанием заняты k каналов;

3) $p_{n+l} = \frac{\rho^{n+l}}{n!n^l} p_0, l = \overline{1, m}$ — вероятность того, что все каналы в СМО заняты и l

требований находятся в очереди;

4) $p_{отк.} = p_{n+m}$ — вероятность отказа;

5) $N_3 = \rho \left[1 - \frac{\rho^{n+m}}{n!n^m} p_0 \right]$ — среднее количество каналов, занятых обслуживанием

(математическое ожидание);

6) $N_{пр.} = n - N_3$ — среднее количество простаивающих каналов;

7) $K_{пр.} = N_{пр.} / n, K_3 = \frac{N_3}{n} = 1 - K_{пр.}$ — коэффициенты простоя и занятости;

8) $q = 1 - p_{отк.}$ — относительная пропускная способность (доля обслуженных требований от общего количества поступивших);

9) $A = \lambda q$ — абсолютная пропускная способность (среднее количество требований, обслуженных системой в единицу времени);

10) $L_{ож.} = \frac{\rho^{n+1}}{n!n} p_0 \left[1 + 2 \frac{\rho}{n} + 3 \left(\frac{\rho}{n} \right)^2 + \dots + m \left(\frac{\rho}{n} \right)^{m-1} \right]$ — среднее количество требований

в очереди;

11) $L = L_{ож.} + N_3$ — среднее количество требований в СМО;

12) $W = \frac{L_{ож.}}{\lambda}$ — среднее время ожидания в очереди;

13) $V = W + \frac{q}{\mu}$ — среднее время пребывания в СМО.

Задание. Рассчитывается пункт обслуживания населения (парикмахерская, химчистка и т.д.). Количество обслуживающих каналов (мастеров, приемщиков) — n с средним временем обслуживания τ . Количество мест в зале ожидания m . Если все места в очереди заняты, то клиент покидает пункт, расчетный поток клиентов для обслуживания случайный, а моменты их прихода распределены по закону Пуассона с параметром λ (клиентов в час).

1) При заданных m, τ, λ рассчитать количество n так, чтобы было обслужено не менее $a\%$ клиентов. Рассчитать для полученного n остальные характеристики СМО.

2) При заданных n, τ, λ рассчитать, максимальное m так, чтобы среднее время ожидания в очереди не превысило b часов.

3) Составить системы уравнений Колмогорова для полученного m .

Исследование операций. Лабораторный практикум.

Составители:

Бышик Татьяна Петровна

Левчук Виктор Дмитриевич

Мережа Валерий Леонидович

Осипенко Наталья Борисовна

Подписано в печать 06.01.98. Формат 60 x 80 1/16. Печать офсетная.
Усл.п.л. 1,6. Уч.-изд.л. 1,3. Тираж 100 экз. Заказ 114 .

Отпечатано на ротапринтере Гомельского государственного университета им.
Ф.Скорины.
246699, Гомель, ул. Советская, 104