

Министерство образования и науки Российской Федерации
Федеральное бюджетное образовательное учреждение
высшего профессионального образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»**
(ВлГУ)

Институт машиностроения и автомобильного транспорта

Кафедра «Автотранспортная и техносферная безопасность»

Методические указания к лекционным занятиям
по дисциплине **«Оптимизационное моделирование транспортных процессов»** для
студентов ВлГУ,
обучающихся по направлению 230301 «Технология транспортных
процессов» профиль «Организация и безопасность движения»

Составитель:

И.В. Денисов

Владимир – 2015 г.

ОГЛАВЛЕНИЕ

1 Моделирование методами сетевого планирования	3
2 Методы динамического моделирования	23
3 Моделирование методами линейного программирования	40
Список литературы	66

1. МОДЕЛИРОВАНИЕ МЕТОДАМИ СЕТЕВОГО ПЛАНИРОВАНИЯ

Сетевое планирование - метод научного планирования и управления производственными процессами, выполняющими большие объемы работ.

Методы сетевого планирования находят широкое применение во многих отраслях народного хозяйства, в том числе и на автомобильном транспорте.

На автомобильном транспорте методами сетевого планирования описываются процессы технического обслуживания и ремонта автомобилей, перевозочные и строительные процессы и т.д.

Сетевое планирование имеет ряд преимуществ:

- обеспечивает наглядность технологической последовательности работ;
- позволяет составить оперативные и текущие планы, а также прогнозировать сложные процессы;
- позволяет выявить скрытые ресурсы времени и материальных средств при выполнении производственных процессов и значительно повысить их эффективность.

Чтобы приступить к сетевому планированию (моделированию) того или иного производственного процесса необходимо иметь перечень, последовательность и продолжительность (трудоемкость) выполнения операций (работ), соответствующих рассматриваемому производственному процессу.

Сетевое планирование сопровождается построением рабочих таблиц и сетевых графиков, к рассмотрению элементов которых мы и перейдем.

1.1. Элементы сетевых графиков

При построении сетевых графиков используют два логических понятия (элемента) – работа и событие.

В сетевом планировании термин «работа» предусматривает процесс предшествующий совершению какого – либо события. Термин «событие» выражает собой определенный результат выполнения работы (или работ).

На сетевом графике события изображают кружком, а работы - ориентированными стрелками.

Фрагмент сетевого графика приведен на рис. 6.1.

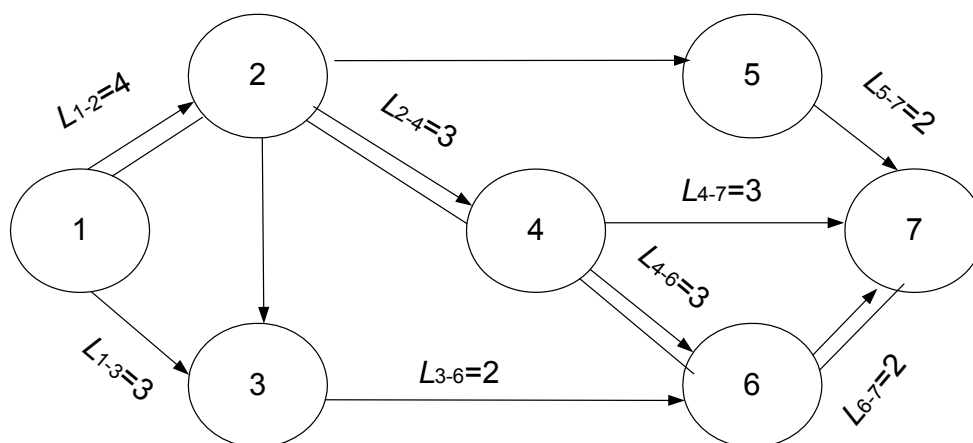


Рис. 1.1

Каждому событию присваивается определенный номер (обычно цифрой), т.е. 1, 2, 3 и т.д. - события.

Каждая работа, изображенная на сетевом графике стрелкой, объединяет только два события, поэтому принято работу на сетевом графике обозначать номерами предшествующего (i -го) и последующего (j -го) событий, т.е. 1 - 2, 2 - 5, 5 - 7 и т.д. - работы.

Продолжительность работы проставляется над стрелками, т. е. $L_{1-2} = 4$, $L_{2-5} = 5$ и т.д. - продолжительность работ.

Сетевой график представляет собой последовательность работ и событий, отражающих их технологическую взаимосвязь.

На сетевом графике выделяют два события: начальное (1) (исходное) и конечное (7) (завершающее). Все остальные события называются промежуточными.

Исходное событие отражает начало выполнения всего комплекса работ и не имеет предшествующего события.

Завершающееся событие отражает конечную цель всего комплекса работ и не имеет последующего события.

Термин «работа» включает три понятия:

1 - «Фактическая работа» - т.е. трудовой процесс, приводящий к достижению определенных результатов и требующих затрат времени и ресурсов;

2 - «Ожидание» - технологический перерыв в работе, не требующий затрат труда, но требующий затрат времени (высыхание краски, отвердевание цемента и т.д.);

3 - «Зависимость» (фиктивная работа) – логическая связь между событиями, не требующая затрат времени и ресурсов, но показывающая, что возможность начала одной работы зависит от результатов другой.

На сетевых графиках фактическую работу и ожидание изображают сплошными стрелками, а зависимости - пунктирными.

Сетевой график строят в масштабе или без масштаба. В последнем случае обязательно над стрелками проставляют продолжительность работы в единицах времени.

Любая последовательность работ от одного события к другому (любому) называется путем и обозначается $L(2 - 5 - 7)$, т. е. каждый путь обозначают буквой L и номерами событий через которые он проходит.

Длина любого пути определяется суммарной продолжительностью составляющих его работ.

Полный путь - это путь от исходного до завершающего события.

В сетевом графике, как правило, имеется несколько полных путей с различной продолжительностью.

Так, для нашего примера имеем пять полных путей, длина которых:

$$L_1(1 - 2 - 5 - 7) = 4 + 5 + 1 = 10;$$

$$L_2(1 - 2 - 4 - 7) = 4 + 3 + 3 = 10;$$

$$L_3(1 - 2 - 4 - 6 - 7) = 4 + 3 + 3 + 2 = 12;$$

$$L_4(1 - 2 - 3 - 6 - 7) = 4 + 1 + 2 + 2 = 9;$$

$$L_5(1 - 3 - 6 - 7) = 4 + 2 + 2 = 8.$$

Полный путь, имеющий максимальную продолжительность, называют критическим путем.

Работы, лежащие на критическом пути называется критическими работами.

Для нашего примера: $L_{кр}(1 - 2 - 4 - 6 - 7) = 12$ единиц времени. Для большей наглядности его выделяют двойными или жирными линиями.

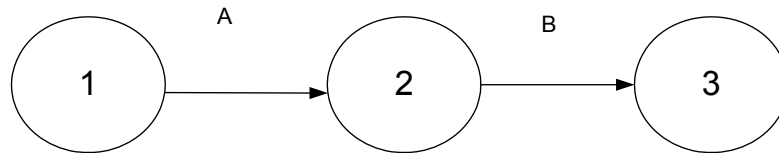
Критический путь ($L_{кр}$) определяет общую продолжительность выполнения всего комплекса работ.

Полные пути, продолжительность которых меньше $L_{кр}$, называются не критическими. У них имеется резерв времени, в пределах которого время выполнения работ может быть увеличено, что не приводит к увеличению общей продолжительности наступления завершающего события.

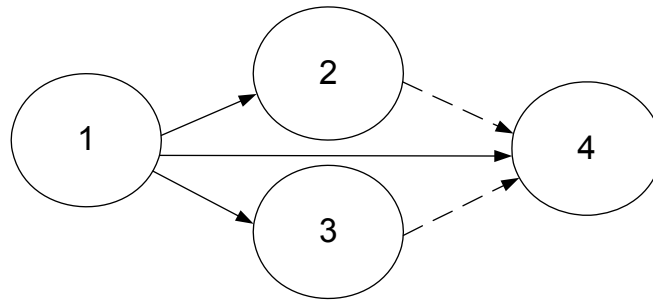
1.2. Правила построения сетевых графиков

При построении сетевых графиков необходимо учитывать следующие правила:

1. Между двумя событиями на графике может находиться только одна работа.

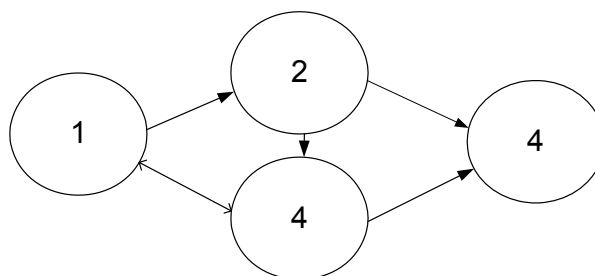


2. Если одно событие служит началом нескольких работ, которые заканчиваются также в одном событии, то необходимо ввести фиктивные работы и дополнительные события со своими номерами.

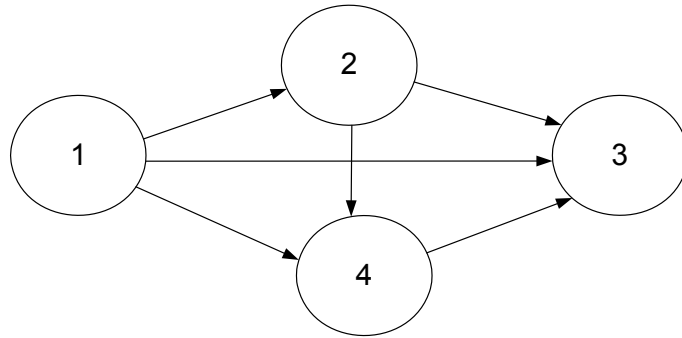


3. Все события, кроме завершающего, должны иметь последующую работу. Наличие «тупиков» в сети указывает на ошибку. Так же в сети не должно быть событий, в которые не входят ни одна работа (исключение составляет исходные событие).

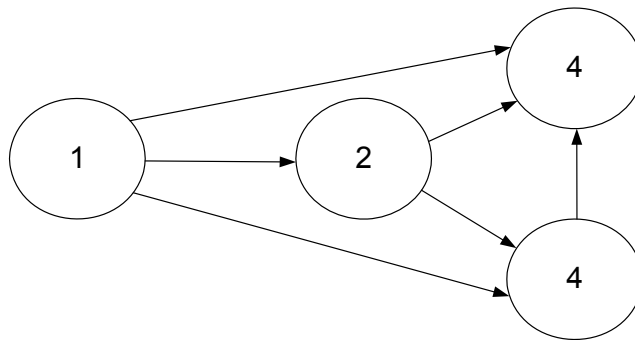
4. В сетевом графике не должно быть замкнутых контуров, т.е. путей, которые начинаются и заканчиваются в одном и том же событии.



5. При построении сетевых графиков следует избегать взаимного пересечения стрелок:



Правильное изображение сети:



1.3. Процесс построения сетевых графиков

Процесс сетевого планирования и управления (СПУ) включает в себя четыре взаимосвязанных этапа:

1. Описание комплекса работ, определение их продолжительности и последовательности.
2. Построение сетевого графика.
3. Расчет и анализ параметров сетевого графика.
4. Оптимизация сетевого графика, контроль и оперативное управление ходом выполнения комплексом работ.

Пример. Составить сетевой график сменно-суточного плана перевозки грузов автомобильным транспортом.

1. Определяем перечень работ в составе сетевого графика (см. табл. 1.1).

Таблица 1.1

№ п/п	Наименование работ	Код работ	Продолжительность, ч
1	Получение заявки на перевозку грузов от клиентуры	1 - 2	7
2	Определение корреспондирующих точек и расстояний между ними	2 - 3	3
3	Определение объемов перевозок	2 - 4	4
4	Выбор подвижного состава для осуществления перевозок	4 - 5	2
5	Фиктивная работа (зависимость)	5 - 6	0
6	Разработка рациональных маршрутов перевозок	6 - 7	1,5
7	Расчет потребного парка подвижного состава	7 - 8	2
8	Составление маршрутных ведомостей (разрядки)	8 - 9	4
9	Выписка путевых листов	9 - 10	3
10	Доставка грузов потребителям	10 - 11	8
11	Контроль за работой подвижного состава на линии	10 - 13	8
12	Прием товарно-транспортных документов	11 - 12	1,5
13	Обработка товарно-транспортных документов	12 - 14	2
14	Составление диспетчерского отчета	14 - 15	2

Примечание: продолжительность работ выбрана условно.

2. Построение сетевого графика.

При этом пользуется основными правилами построения сетевых графиков, а также учитываем, что взаимосвязанные работы выполняются последовательно, независимые работы выполняются как последовательно, так и параллельно.

Строим сетевой график сменно-суточного плана перевозки грузов (см. рис. 1.2.).

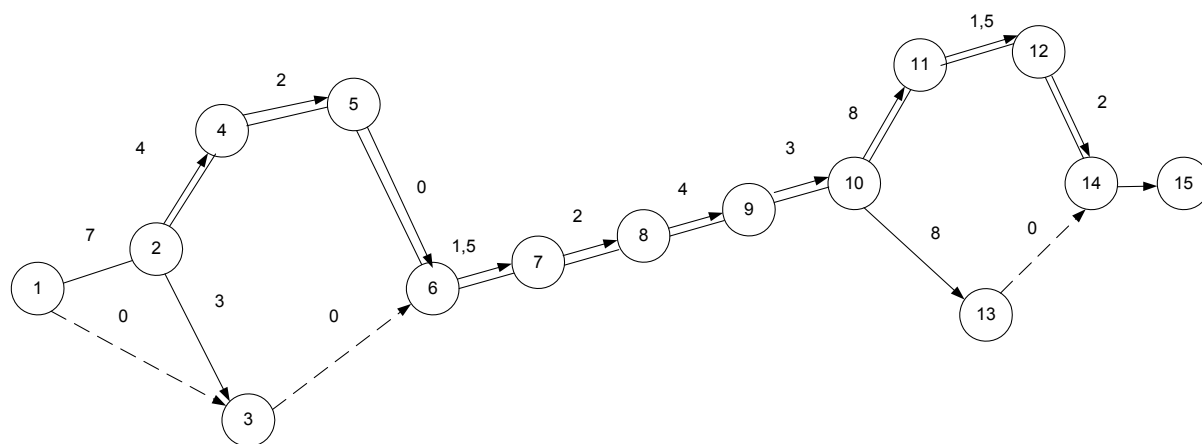


Рис. 1.2. Сетевой график сменно – суточного плана перевозки грузов

При построении сетевого графика использованы следующие основные правила:

1. События обозначают кружочками, внутри ставится номер события (выделяются начальное, конечное и промежуточные события).
2. События соединяются ориентированной стрелкой, которая направлена от предшествующего события к последующему (стрелка представляет собой на сетевом графике работу).
3. Любые два события могут быть соединены не более чем одной стрелкой.
4. В начальное событие не входит ни одна стрелка.
5. Из конечного события не выходит ни одна стрелка.
6. События сетевого графика нумеруются так, чтобы для каждой работы номер начального события был меньше, чем номер конечного.
7. Каждую работу кодируют двумя цифрами. Первая цифра означает начало работы и соответствует номеру предшествующего события.
8. Продолжительность работы проставляется над стрелками.
9. Часть работ выполняется последовательно (6 – 7), (7 – 8), (8 – 9) и т.д. Это означает, что начало каждой последующей работы зависит от окончания предшествующей.
10. Работы (10 – 11), (10 – 13) могут начинаться в один и тот же момент времени с наступлением события 10. Эти работы не зависят во времени одна от другой и могут выполняться параллельно.
11. Фиктивные работы (1 – 3), (3 – 6) и т.д. устанавливают логическую взаимосвязь и продолжительность их равна 0.
12. Весь комплекс работ завершается, как только окончится работа (14 – 15) и свершится событие 15.

Составление сетевого графика сменно-суточного плана перевозок этим считается законченным.

При разработке сетевых графиков необходимо учитывать следующие условия:

1. Ни одно условие не может произойти до тех пор, пока не будет заключены все входящие в него работы.
2. Ни одна работа, выходящая из данного события, не может начинаться до тех пор, пока данное событие не произойдет.

На сетевом графике выдается критический путь для последующей его минимизации. Для чего определяют время начала и окончания каждой операции, время наступления каждого события, а также устанавливают возможность изменения этих параметров с целью оптимизации сетевой модели.

1.4. Расчет параметров сетевой модели

Расчет параметров сетевой модели ведется для полных путей, событий и работ.

При расчетах определяют следующие параметры:

- А) для полных путей сетевого графика:
 - $t(L_i)$ - продолжительность любого полного пути;
 - $t(L_{кр})$ - продолжительность критического пути;
 - $R(L_i)$ - полный резерв времени пути.
- Б) для событий:
 - $T_i^{(p)}, T_i^{(n)}$ - ранний и поздний сроки совершения события;
 - R_i - резерв времени события.
- В) для работ:
 - $t_{ij}^{(pn)}, t_{ij}^{(po)}$ - ранний срок начала и окончания работ;
 - $t_{ij}^{(nn)}, t_{ij}^{(no)}$ - поздний срок начала и окончания работ;
 - $r_{ij}^{(n)}, r_{ij}^{(св)}$ - полный и свободный резерв времени работы.

При расчете этих параметров используют графический и табличный методы.

1.4.1. Расчет продолжительности полного пути

1. Расчет продолжительности любого полного пути осуществляется по формуле

$$t(L_i) = \sum^{L_i} t_{ij}$$

Расчет сведем таблицу 1.2.

Таблица 1.2

№ п/п	Путь, L_i	Продолжительность пути, $t(L_i)$	Резерв, $R(L_i)$	Примечание
1	1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15	$7 + 4 + 2 + 0 + 1,5 + 2 + 4 + 3 + 8 + 1,5 + 2 + 2 = 37$	0	Критический путь
2	1 – 2 – 4 – 5 – 6 – 2 – 8 – 9 – 10 – 13 – 14 – 15	$7 + 4 + 2 + 0 + 1,5 + 2 + 4 + 3 + 8 + 0 + 2 = 33,5$	3,5	
3	1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15	$7 + 3 + 0 + 1,5 + 2 + 4 + 3 + 8 + 1,5 + 2 + 2 = 34$	3,0	
4	1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 13 – 14 – 15	$7 + 3 + 0 + 1,5 + 2 + 4 + 3 + 8 + 0 + 2 = 30,5$	4,5	
5	1 – 3 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 14 – 15	$0 + 0 + 1,5 + 2 + 4 + 3 + 8 + 1,5 + 2 + 2 = 24$	13,0	
6	1 – 3 – 6 – 7 – 8 – 9 – 10 – 13 – 14 – 15	$0 + 0 + 1,5 + 2 + 4 + 3 + 8 + 0 + 2 = 20,5$	16,5	

2. $t(L_h) = \max \sum^{L_i} t_{ij}$ - продолжительность критического пути (в нашем примере это путь 1).

3. Полный резерв времени пути

$$R(L_i) = t(L_{кр}) - t(L_i).$$

Повышение суммарной продолжительности всех работ, лежащих на пути L_i , на величину $R(L_i)$ не увеличивает время наступления завершающего события.

1.4.2. Расчет времени наступления событий

Расчет произведем на сетевом графике рис. 6.3.

При графическом методе записи расчетных параметров осуществляется непосредственно на сетевом графике.

Для чего каждый кружок сетевого графика делим на четыре части (секторы), в этих секторах записываются следующие данные:

- верхний - предназначен для записи номера события - i ;
- правый - для записи раннего срока свершения события - T_i^P ;
- левый - для записи позднего срока свершения события - T_i^N ;
- нижний - для записи резерва времени события - R_i .

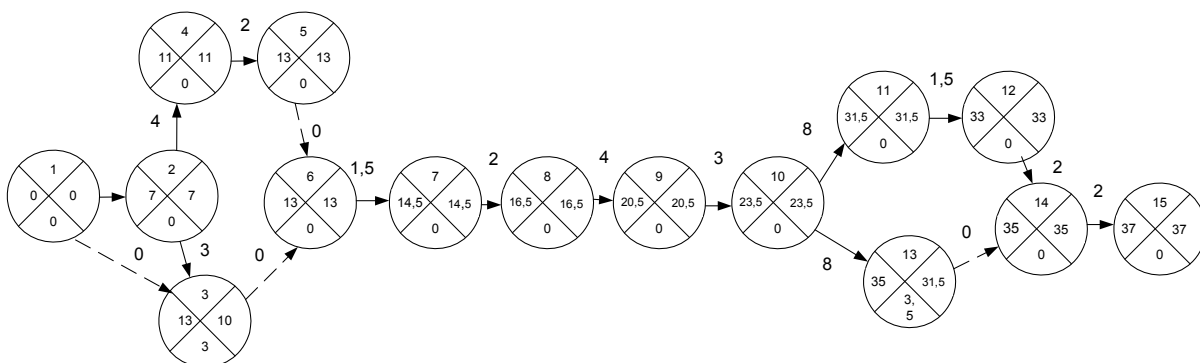


Рис. 1.3. Сетевой график с указанием времени наступления событий

1. Наиболее ранний срок поступления i -го события в сети T_i^P , где $i = 1, 2, \dots, n$; i - одно из событий сети.

T_i^P - минимально необходимое время между наступлением начального и данного события.

Для начального события $T_1^P = 0$ - наиболее ранний срок равен 0. При расчете T_i^P последовательно переходят от начального события к событию, все более от него удаленному. Тогда для любого другого события j этот показатель определяется по формуле

$$T_j^P = \max [T_i^P + t_{ij}],$$

где $T_i^{(p)}$ - наиболее ранний срок поступления события i , предшествующего событию j ;

t_{ij} - продолжительность работы ($i - j$).

Для конечного события сетевого графика наиболее ранний срок наступления его равен продолжительности критического пути и называется критическим временем сетевого графика.

2. Наиболее поздний срок наступления события в сети T_i^n .

Этот показатель рассчитывается от конца сетевого графика к началу, т.е. в направлении, обратном определению наиболее раннего срока наступления событий.

Для конечного события (k) делается предположение, что наиболее ранний срок его наступления равен наиболее позднему сроку, т.е.

$$T_k^p = T_k^n.$$

Для критического пути также верно равенство

$$T_{кр}^p = T_{кр}^n.$$

Тогда для начального - $T_1^n = 0$.

Для других событий сетевого графика T_i^n определяется по формуле

$$T_i^n = \min [T_j^n - t_{ij}],$$

где T_j^n - наиболее поздний срок наступления последующего события j ;

t_{ij} - продолжительность работы ($i - j$).

Этот показатель определяет наиболее допустимое время наступления события, не требующее увеличения времени на осуществление всего комплекса работ.

Допустимый срок наступления события - T_j^d

$$T_j^p \leq T_j^d \leq T_j^n.$$

Данное неравенство показывает, что допустимый срок наступления события должны находиться в диапазоне изменений от наиболее раннего срока наступления до наиболее позднего срока наступления данного события.

Для критических событий

$$T_{кр}^p = T_{кр}^d = T_{кр}^n.$$

3. Резерв времени событий - R_i .

Рассчитав ранние и поздние сроки наступления каждого события, можно определить резервы времени событий по формуле

$$R_i = T_i^n - T_i^p.$$

Резервы времени всего критических событий равны 0.

$$R_{икр} = 0.$$

1.4.3. Расчет времени выполнения работ

Расчет времени выполнения работ проводят после того, как определены T_i^p и T_i^n для всех событий:

а) Ранний срок начала работ ($t_{ij}^{p.H}$) равен раннему сроку наступления события, из которого исходит данная работа, т.е.

$$t_{ij}^{p.H} = T_i^p.$$

Если эту оценку выразить через характеристики работ, то можем записать

$$t_{jk}^{p.H} = t_{ij}^{p.H} + t_{ij},$$

где t_{ij} - предшествующая работа; t_{jk} - последующая работа.

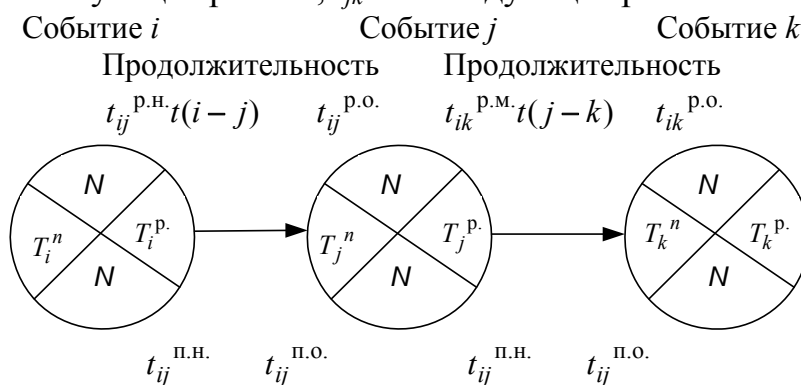


Рис. 1.4. Фрагмент сетевого графика

б) Ранний срок окончания работы определяется путем прибавления к раннему сроку начала работы продолжительности самой работы

$$t_{ij}^{p.o} = T_i^p + t_{ij} \quad \text{или} \quad t_{ij}^{p.o} = t_{ij}^{p.H} + t_{ij}.$$

в) Поздний срок окончания работы равен позднему сроку наступления последующего события

$$t_{ij}^{п.о} = T_j^n \quad \text{или} \quad t_{ij}^{п.о} = t_{jk}^{п.о} - t_{ij}.$$

г) Поздний срок начала работы находится путем вычитания из позднего срока наступления последующего события продолжительности работы, т.е.

$$t_{ij}^{п.н} = T_j^n - t_{ij} \quad \text{или} \quad t_{ij}^{п.н} = t_{jk}^{п.о} - t_{ij}.$$

д) Полный резерв времени работы показывает время, на которое можно перенести начало данной работы (или увеличить её продолжительность), не изменяя при этом длины критического пути и определяется по формулам

$$r_{ij}^n = t_{ij}^{п.н} - t_{ij}^{p.H} \quad \text{или} \quad r_{ij}^n = T_j^n - T_i^p - t_{ij}.$$

Для всех работ, лежащих на критическом пути

$$r_{кр ij}^n = 0.$$

е) Свободный резерв времени работы - часть полного резерва времени работы, которая сохраняется у нее при условии, что начальное событие работы совершится в самый поздний срок, а конечное - в самый ранний срок и определяется по формулам

$$r_{ij}^{CB} = r_{ij}^n - R_i - R_j \text{ или } r_{ij}^{CB} = T_j^P - T_i^H - t_{ij}.$$

Результаты расчета временных характеристик работ рассматриваемого сетевого графика сведены в табл. 1.3.

Таблица 1.3.

Таблица расчета временных характеристик сетевого графика

№ п/п	Код работы	Продолжительность работ	Начало работ		Окончание работ		Запас времени полный
			раннее	позднее	раннее	позднее	
1	1 - 2	7	0	0	7	7	0
2	2 - 3	3	7	10	10	13	3
3	2 - 4	4	7	7	11	11	0
4	4 - 5	2	11	11	13	13	0
5	5 - 6	0	13	13	13	13	0
6	6 - 7	1,5	13	13	14,5	14,5	0
7	7 - 8	2	14,5	14,5	16,5	16,5	0
8	8 - 9	4	16,5	16,5	20,5	20,5	0
9	9 - 10	3	20,5	20,5	23,5	23,5	0
10	10 - 11	8	23,5	23,5	31,5	31,5	0
11	10 - 13	8	23,5	27	31,5	35	3,5
12	11 - 12	1,5	31,5	31,5	33	33	0
13	12 - 14	2	33	33	35	35	0
14	14 - 15	2	35	35	32,0	37	0

1.5. Сетевой график ремонта автомобилей

Дополнительно принципы построения сетевых графиков рассмотрим на основе перечня работ ремонта автомобиля и времени их выполнения (см. табл. 1.4.)

В табл. 1.4. указано, что выполнение работ a_3 , a_4 и a_5 может начаться только после окончания работы a_2 ; работы a_6 - после окончания a_3 и т.д. Установление очередности и полного перечня работ - очень важный вопрос при составлении исходных данных для сетевой модели и исследуемого процесса.

Используя данные табл. 1.4, можно построить сетевой график (см. рис. 1.5.), где работы изображены векторами (стрелками), а события, состоящие в окончании работ и возможности начать новые работы, - кружками.

Чтобы отразить на графике зависимость непосредственно не опирающихся друг на друга работ, вводится фиктивная работа, имеющая нулевые затраты времени для ее выполнения. Она наносится на график пунктирной линией и нужна лишь для того, чтобы изобразить на сети требуемую очередность работы. Например, на рис. 1.5 событие 11 (начало работы a_{12}) возможно только тогда, когда будут выполнены работы a_9 и a_{10} , связанные через фиктивные работы (9 - 11) и (10 - 11).

Таблица 1.4

№ п/п	Вид работ	Условное обозначение вида работ	Предшествующий вид работ	Продолжительность работ, ч
1	Подготовка автомобиля к ремонту и постановка его на место разборки	a_1	-	2
2	Разборка автомобиля на основные узлы	a_2	a_1	8
3	Разборка и дефектовка деталей двигателя	a_3	a_2	6
4	Разборка и дефектовка деталей трансмиссии	a_4	a_2	6
5	Разборка и дефектовка деталей рулевого управления и ходовой части	a_5	a_2	4
6	Комплектовка деталей двигателя	a_6	a_3	4
7	Комплектовка деталей трансмиссии	a_7	a_4	4
8	Комплектовка деталей ходовой части и рулевого управления	a_8	a_5	3
9	Сборка и испытания двигателя	a_9	a_6	12
10	Сборка и испытание ходовой части и рулевого управления	a_{10}	a_8	10
11	Сборка и испытание трансмиссии	a_{11}	a_7	14
12	Сборка автомобилей из узлов и обкатка	a_{12}	a_9, a_{10}, a_{11}	34
13	Покраска	a_{13}	a_{12}	2
14	Приемка автомобиля из ремонта	a_{14}	a_{13}	0,5

Часть работ выполняется последовательно (2 - 3), (3 - 6), (6 - 9) и т. д. Это означает, что начало каждой последующей работы зависит от окончания предшествующей.

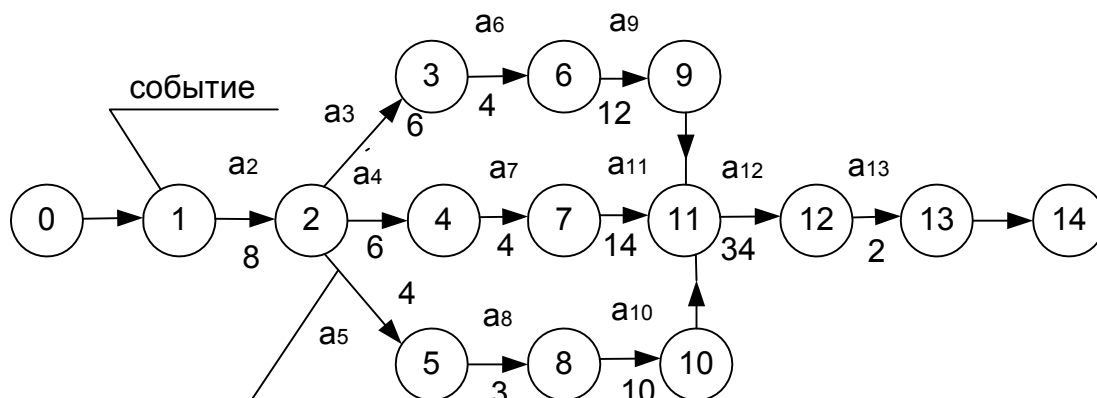


Рис. 1.5. Упрощенный сетевой график ремонта автомобиля

Работы (2 - 3), (2 - 4), (2 - 5) могут начинаться в один и тот же момент времени с наступлением события 2. Эти работы не зависят во времени одна от другой и могут выполняться параллельно.

Фиктивные работы (9 - 11), (10 - 11) устанавливают логическую взаимосвязь, и продолжительность их равна 0.

После построения сетевого графика производится расчет параметров модели: продолжительность критического пути, сроков наступления событий и времени выполнения работ.

Критический путь - это последовательность технологически взаимосвязанных работ от начального до конечного событий, имеющая максимальную продолжительность. Найдем критический путь для рассмотренного примера, определив, время выполнения работ по каждой из трех ветвей:

верхняя ($a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_6 \rightarrow a_9 \rightarrow a_{12} \rightarrow a_{13} \rightarrow a_{14}$)

$$T_B = 2 + 8 + 6 + 4 + 12 + 34 + 2 + 0,5 = 68,5 \text{ (ч)};$$

средняя ($a_1 \rightarrow a_2 \rightarrow a_4 \rightarrow a_7 \rightarrow a_{11} \rightarrow a_{12} \rightarrow a_{13} \rightarrow a_{14}$)

$$T_C = 2 + 8 + 6 + 4 + 14 + 34 + 2 + 0,5 = 70,5 \text{ (ч)};$$

нижняя ($a_1 \rightarrow a_2 \rightarrow a_5 \rightarrow a_8 \rightarrow a_{10} \rightarrow a_{12} \rightarrow a_{13} \rightarrow a_{14}$)

$$T_H = 2 + 8 + 4 + 3 + 10 + 34 + 2 + 0,5 = 63,5 \text{ (ч)}.$$

Наибольшее время выполнения работ мы получили на средней ветви графика, этот путь и является критическим

$$T_{кр} = T_C = 70,5 \text{ (ч)}.$$

Работам, лежащим на критическом пути, уделяется особое внимание, поскольку всякая задержка в выполнении любой из этих работ приводит к срыву окончания всего комплекса работ.

Расчет сроков наступления событий ведется в следующей последовательности.

Первоначально определяется наиболее ранний срок наступления j -го события в сети $T_p(j)$, где $j = 1, 2, \dots, n$; n - одно из событий сети. Для начального события наиболее ранний срок равен 0, т.е. $T_p(1) = 0$. Для любого другого события этот показатель определяется по формуле

$$T_p(j) = \max[T_p(i) + t_{ij}],$$

где $T_p(i)$ - наиболее ранний срок наступления события i , предшествующего j , t_{ij} - продолжительность работы $(i - j)$ (рис. 1.6.).

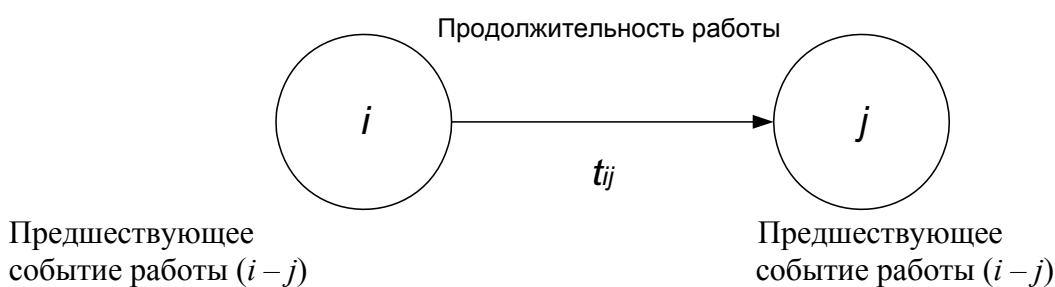


Рис. 1.6. Элементы сетевого графика

Для конечного события сетевого графика наиболее ранний срок наступления равен продолжительности критического пути.

Наиболее поздний срок наступления события в сети $T_n(i)$ определяется по формуле

$$T_n(i) = \min[T_n(j) - t_{ij}],$$

где $T_n(j)$ - наиболее поздний срок наступления последующего события j ; t_{ij} - продолжительность работы $i - j$.

Этот показатель рассчитывается от конца сетевого графика к началу, т.е. в направлении, обратном определению наиболее раннего срока наступления событий. Для конечного события k делается предположение, что наиболее ранний срок его наступления равен наиболее позднему сроку, т.е. $T_p(k) = T_n(k)$. Для критических событий сетевого графика $T_p(i) = T_n(i)$. Для начального события $T_n(1) = 0$.

Допустимый срок наступления события $T_a(j)$ $T_p(j) \leq T_a(j) \leq T_n(j)$.

Данное неравенство показывает, что допустимый срок наступления события должен находиться в диапазоне изменений от наиболее раннего срока наступления до наиболее позднего допустимого срока наступления данного события. Для критических событий

$$T_p(i) = T_a(i) = T_n(i).$$

Результаты расчетов сроков наступления событий для сетевого графика, представленного на рис. 1.5, приведены в табл. 1.5.

Таблица 1.5

№ п/п	Событие	Срок наступления	
		Наиболее ранний, $T_p(i)$	Наиболее поздний, $T_n(i)$
0	Автомобиль поставлен на ремонт	0	0*
1	Начата разборка автомобиля на узлы	2	2*
2	Начата разборка узлов автомобиля	10	10*
3	Закончена разборка и дефектовка деталей двигателя	16	18
4	Закончена разборка и дефектовка деталей трансмиссии	16	16*
5	Закончена разборка и дефектовка ходовой части и рулевого управления	14	21
6	Начата сборка двигателя	20	22
7	Начата сборка трансмиссии	20	20*
8	Начата сборка ходовой части и рулевого управления	17	24
9	Закончена сборка и испытание двигателя	32	34
10	Закончена сборка ходовой части и рулевого управления	27	34
11	Закончена сборка трансмиссии и начата сборка автомобиля из узлов	34	34*
12	Закончена сборка автомобиля	68	68*
13	Закончена покраска автомобиля	70	70*
14	Автомобиль принят с ремонта	70,5	70,5*

* События, лежащие на критическом пути.

В рамках сетевого графика каждая работа характеризуется следующими четырьмя временными параметрами:

1) Раннее начало работы - $t_{ij}^{p.H}$ определяется как продолжительность пути наибольшей длины от начального события до предшествующего события данной работы. Раннее начало любой последующей работы ($j - k$) равно сумме значений раннего начала и продолжительности работы ($i - j$), предшествующей для данной работы (рис. 6.4)

$$t_{kj}^{p.H} = t_{ij}^{p.H} + t_{ij}.$$

Раннее начало работ, выходящих из первого события, равно нулю. Если данной работе предшествуют две и большее количество работ, то ее раннее

начало равно максимальному значению сумм раннего начала и продолжительности предшествующей работы $t_{jk}^{p.H} = \max[t_{ij}^{p.H} + t_{ij}]$.

Раннее начало работы $j - k$ равно наиболее раннему сроку наступления предшествующего события j , т.е. $t_{jk}^{p.H} = T_p(j)$;

2) раннее окончание работы $t_{ij}^{p.O}$ определяется как сумма раннего начала работ и ее продолжительности $t_{ij}^{p.O} = t_{ij}^{p.H} + t_{ij}$;

3) позднее начало работ $t_{ij}^{n.O}$ представляет собой самый поздний срок начала работы, который не вызывает задержки выполнения всего комплекса в целом. Позднее начало работы рассчитывается в обратном порядке, от конца сетевого графика к началу, и определяется как разность между продолжительностью критического пути от конечного состояния графа до предшествующего события данной работы;

4) позднее окончание работы $t_{ij}^{n.O}$ равно времени окончания работы, если она была начата в поздний срок, и поэтому определяется как сумма, позднего начала и ее продолжительности $t_{ij}^{n.O} = t_{ij}^{n.H} + t_{ij}$.

Если известно позднее окончание последующей работы, то для данной работы значение позднего окончания определится следующим образом:

$$t_{ij}^{n.O} = t_{jk}^{n.O} - t_{jk}.$$

Позднее окончание работы $t_{ij}^{n.O}$ сетевого графика всегда равно наиболее позднему сроку наступления последующего события $T_{п}(j)$, т.е. $t_{ij}^{n.O} = T_{п}(j)$.

Результаты счета временных параметров работ сведены в табл. 6.6.

Из табл. 6.6. следует, что значение критического пути определяется по максимальной величине их ранних окончаний работ $t_{ij}^{p.O}$ и равно 70,5.

Позднее окончание работы (13 - 14), завершающееся событием 14, равно максимальному значению их ранних окончаний работ и также определяет продолжительность критического пути.

Работы, у которых ранние начала и окончания соответственно равны поздним началам и окончаниям, лежат на критическом пути и не имеют запасов времени. Для данного графика к этим работам относятся (0 - 1), (1 - 2), (2 - 4), (4 - 7), (7 - 11), (11 - 12), (12 - 13), (13 - 14).

Полный (общий) запас времени r_{ij}^n , представляющий собой время, на которое можно перенести начало работ ($i - j$) или, наоборот, увеличить их продолжительность без изменения общего срока выполнения комплекса работ, и определен по формуле $r_{ij}^n = t_{ij}^{n.H} - t_{ij}^{p.H}$.

Таблица 1.6

Кол-во работ предшествующих данной работе	Код работы	Продолжительность работы, ч.	Начало работы		Окончание работы		Запас времени
			раннее	позднее	раннее	позднее	полный
			$t_{ij}^{p.n}$	$t_{ij}^{п.н}$	$t_{ij}^{p.o}$	$t_{ij}^{п.o}$	$r_{ij}^{п}$
0	0-1	2	0	0	2	2	0
1	1-2	8	2	2	10	10	0
1	2-3	6	10	12	16	18	2
1	2-4	6	10	10	16	16	0
1	2-5	4	10	12	14	21	7
1	3-6	4	16	18	20	22	2
1	4-7	4	16	16	20	20	0
1	5-8	3	14	21	17	24	7
1	6-9	12	20	22	32	34	2
1	7-11	14	20	20	34	34	0
1	8-10	10	17	24	27	34	7
1	9-11	0	32	34	32	34	2
1	10-11	0	27	34	27	34	7
3	11-12	34	34	34	68	68	0
1	12-13	2	68	68	70	70	0
1	13-14	0,5	70	70	70,5	70,5	0

1.6. Оптимизация и преимущества сетевых моделей

После того, как построен исходный сетевой график и рассчитаны основные параметры сетевой модели, необходимо дать оценку полученным результатам. Если критический путь больше установленного срока, предложенного руководством, то необходимо осуществить оптимизацию сетевого графика.

Под оптимизацией понимают процесс улучшения сетевого графика путем:

- а) уменьшения общего времени выполнения работ критического пути (оптимизация по времени);
- б) обеспечение выполнения всего комплекса работ при ограниченных ресурсах (оптимизация по ресурсам).

Оптимизация сетевого графика по времени заключается в сокращении критического пути и проводится в следующем порядке:

1. уточнение времени выполнения работ;

2. сокращение времени выполнения критических работ за счет совершенствования технологии их производства;
3. изучаются возможности замены последовательно выполняемых работ параллельными там, где это допускается технологией, с целью сокращения продолжительности работ;
4. перераспределение ресурсов с некритических на критические работы, выполнение последних в две или три смены;
5. сокращаются сроки выполнения комплекса работ за счет привлечения дополнительных ресурсов, а также применения технологических условий производства комплекса работ.

Если исходный вариант сетевого графика имеет продолжительность критического пути, соответствующую директивному сроку или не превышает этот срок, то он считается оптимальным и может быть рекомендован к утверждению и выполнению.

Оптимизация сети по ресурсам. Оптимизация сетевого графика по времени без учета ограничений по ресурсам предполагает, что потребность в ресурсах может быть установлена в необходимые сроки. Однако такой подход к разработке сетевых графиков не исключает решения задачи наиболее рационального распределения ресурсов, поэтому после оптимизации сетевой модели по критерию «времени» производится ее оптимизация по ресурсам.

Преимуществами сетевых моделей являются:

1. Сетевые графики дают четкое представление об общем объеме работ комплекса.
2. Обеспечивают наглядность технологической последовательности работ.
3. Сетевые модели обеспечивают распределение средств и рабочей силы, что создает условия для наилучшего использования ресурсов.
4. Позволяют осуществлять текущее планирование выполнения работ, прогнозировать сложные процессы, выявлять «узкие места» производства.
5. Сократить потери времени при выполнении всего комплекса работ.
6. Выбрать оптимальный вариант выполнения работ.

Контрольные вопросы

1. Назначение сетевого планирования.
2. Элементы сетевых графиков и их отображение на сетевой модели.
3. Что такое «критический путь»?
4. Перечислите основные правила построения сетевых графиков.
5. Перечислите этапы построения сетевых графиков.

6. Параметры сетевых моделей для полного пути и способы их вычисления.
7. Параметры сетевых моделей для событий и способы их вычисления.
8. Параметры сетевых моделей для работ и способы их вычисления.
9. Допустимый срок наступления события и резерв времени события.
10. Полный и свободный резерв времени работы и способы их вычисления.
11. Сущность оптимизации сетевого графика по времени и по ресурсам.
12. Преимущества сетевых моделей.

2. МОДЕЛИРОВАНИЕ МЕТОДАМИ ДИНАМИЧЕСКОГО МОДЕЛИРОВАНИЯ

Динамическое программирование - особый метод решения оптимизационных задач. Особенностью данного метода является то, что для нахождения оптимального решения задача разбивается на этапы (шаги) и оптимальное решение отыскивается постепенно шаг за шагом.

Методом динамического программирования решаются следующие задачи автомобильного транспорта:

- задачи маршрутизации;
- задачи замены оборудования и подвижного состава;
- оптимизация управления запчастями;
- оптимизация распределения ресурсов и др.

2.1. Общая постановка задачи динамического программирования

Пусть имеем некоторую физическую систему (автомобиль), которая с течением времени может менять свое состояние (процесс старения автомобиля и т.п.), т. е. в системе происходит какой-то процесс. Поставим задачу управлять этим процессом.

Итак, система S (автомобиль) может из начального состояния S_0 перейти в конечное состояние S_m , но не просто, а под действием некоторого управления U (рис. 2.1.)

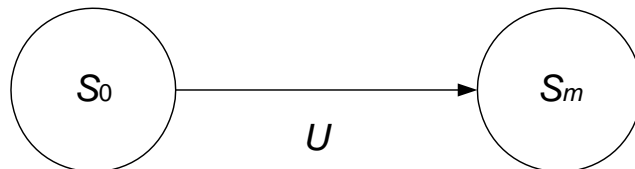


Рис. 2.1. Схема состояний системы

Управление должно быть таким, чтобы оно дало некоторый «выигрыш», который обозначим через W .

Этот выигрыш зависит от управления, т.е. можем записать

$$W = f(U).$$

Очевидно, что мы должны найти такое управление, при котором выигрыш будет максимальным, т.е.

$$W_{\max} = \max_u \{f(u)\},$$

где u - возможные управления;

u - оптимальное управление;

\max_u - «максимум по u , т.е. максимальное значение $f(u)$ при всех возможных управлениях U .

Итак, общую задачу динамического программирования можно сформулировать так:

Из множества возможных управлений U надо найти такое оптимальное управление U , которое переводит физическую систему S из начального состояния S_0 в конечное состояние S_m так, чтобы при этом выигрыш W обращается в максимум.

2.2. Принципы оптимизации задач динамического программирования

Принципы оптимизации сводятся к следующему:

а) Процесс перемещения системы из состояния S_0 в состояние S_m разбивается искусственно или естественно на несколько шагов (этапов) (рис. 2.2. и 2.3.).

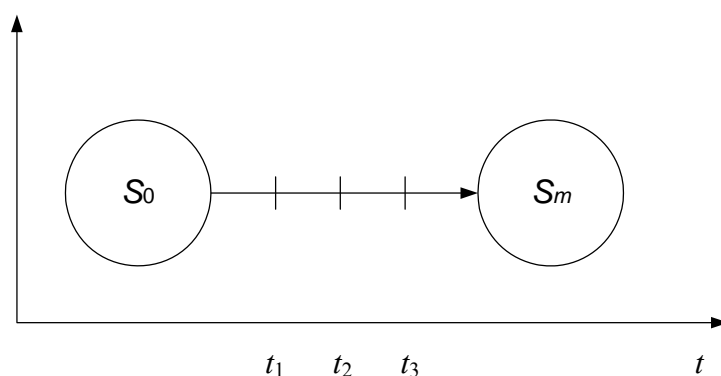


Рис. 2.2

В данном случае имеем m шагов.

б) Производится пошаговая оптимизация, заключающаяся в получении выигрыша на каждом шаге.

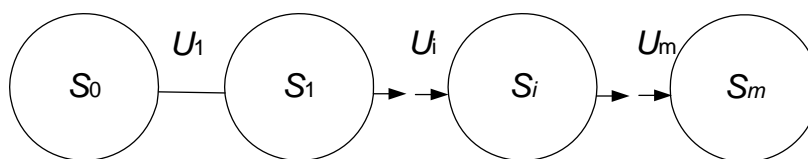


Рис. 2.3

Если выберем на первом шаге управление U_1 , то выигрыш на этом шаге от принятого управления составит

$$W_1 = f(S_0, U_1).$$

Для i -го шага с управлением U_i , выигрыш составит

$$W_i = f(S_{i-1}, U_i) \text{ и т.д.}$$

Т.е. выигрыш на i -ом шаге есть функция состояния и принятого управления.

Процедура построения оптимального управления включает две стадии:

1. предварительную (условную);

2. окончательную (безусловную).

Предварительная (условная) оптимизация производится по шагам в обратном порядке – от последнего шага к первому. На предварительной стадии определяется для каждого шага условное оптимальное управление и условный оптимальный выигрыш.

Окончательная (безусловная) оптимизация производится также по шагам, но в естественном порядке - от первого шага к последнему.

На окончательной стадии определяется для каждого шага окончательное (безусловное) оптимальное управление и безусловный оптимальный выигрыш.

В основе процедуры оптимизации задач динамического программирования лежит уравнение Беллмана, к рассмотрению которого и перейдем.

2.3. Основное уравнение динамического программирования (уравнение Беллмана)

Пусть имеем задачу динамического программирования, рассматривающую процесс перехода системы S из состояния S_0 в состояние S_m .

Для решения данной задачи процесс перехода системы из состояния S_0 в состояние S_m разбиваем на m шагов. Получаем (рис. 2.4.).

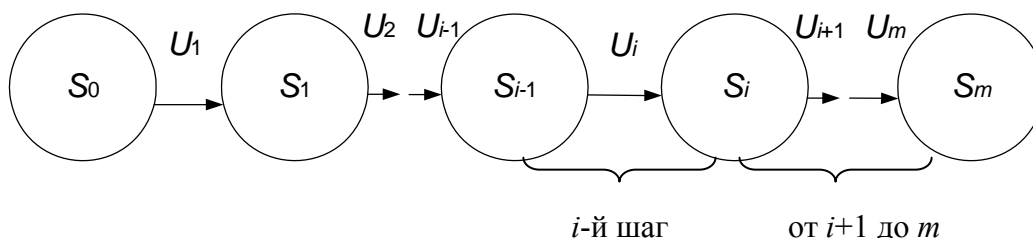


Рис. 2.4

Будем управлять этим процессом, т.е. принимать управления $U_1 \rightarrow U_2 \rightarrow$ и т.д.

1. Выберем на i -ом шаге управление U_i , которое дает на этом шаге выигрыш W_i

$$W_i = f(S_{i-1}, U_i) = W_i(S_{i-1}, U_i).$$

Подобный выигрыш мы можем иметь на каждом шаге от 1 до m .

2. Обозначим оптимальный общий выигрыш, получаемый на всех шагах следующих за i через

$$W_{i+1}^*(S_i)$$

- общий выигрыш на шагах от $(i + 1)$, ... , до m , который выбирается оптимальным.

3. Таким образом, общий выигрыш, который мы имеем на всех шагах, начиная с i - го, можем представить формулой

$$W_i(S_{i-1}) = w_i(S_{i-1}, U_i) + W_{i+1}^*(S_i)$$

т.е. общий выигрыш равен сумме выигрышей:

- выигрыш на i -ом шаге - w_i ;
- и оптимальный выигрыш на всех последующих шагах, начиная с $i + 1$ до m - $W_{i+1}^*(S_i)$.

4. В соответствии с принципом оптимизации, мы должны выбирать на i -ом шаге такое управление $U_i = u_i$, при котором выигрыш был бы максимальным, т.е.

$$W_i^*(S_{i-1}) = \max_{U_i} \{w_i(S_{i-1}, U_i) + W_{i+1}^*(S_i)\} \quad (2.1)$$

Выражение (7.1) характеризует условный оптимальный выигрыш на всех шагах с i -го до m (до конца) и называется рекуррентным уравнением Беллмана.

Вычисление составляющих выражения (2.1) начинают с последнего m -го шага.

2.3.1. Предварительная (условная) оптимизация

Используя уравнение (2.1) определим условный оптимальный выигрыш на последнем шаге m

$$W_m^*(S_{m-1}) = \max \{W_m(S_{m-1}, U_m)\} \quad (2.2)$$

Отметим, что последнее слагаемое в формуле (7.1) равно 0, т.к. за S_m нет другого состояния.

Выражение (2.2) определяет условный оптимальный выигрыш на последнем шаге, который достигается при управлении

$$U_m = u_m(S_{m-1}).$$

Схематические соотношения (2.1) и (2.2) можем проиллюстрировать рисунком 2.5.

Итак, зная выигрыш на m шаге, можем найти выигрыш на шаге $m - 1$ для чего используем рекуррентную формулу (2.1), т.е. используя выражение (7.1), можем построить всю цепочку условных оптимальных управлений и условных оптимальных выигрышей.

Действительно, зная $W_m^*(S_{m-1})$ можно по рекуррентному уравнению Беллмана (2.1) найти $W_{m-1}^*(S_{m-2})$ и $u_{m-1}(S_{m-2})$, а затем $W_{m-2}^*(S_{m-3})$ и $u_{m-2}(S_{m-3})$ и т. д. до последнего от конца (первого) шага.

$$W_1^*(S_1) \text{ и } U_1(S_0).$$

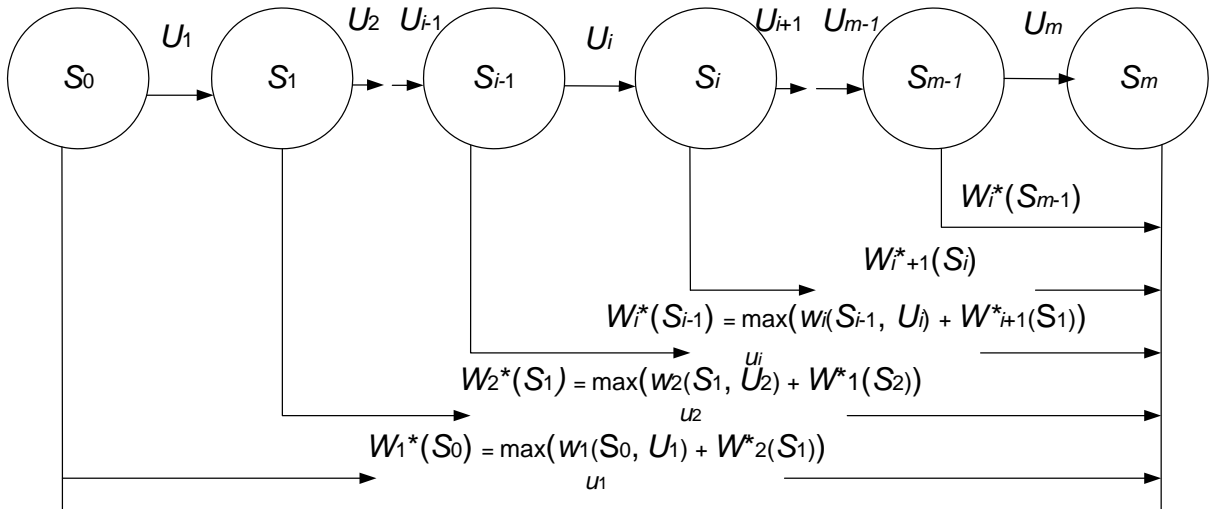


Рис. 2.5

Функция $W_1^*(S_0)$ - есть условный оптимальный выигрыш за всю операцию, т.е. на всех шагах, начиная с последнего, и до первого.

На этом предварительная оптимизация заканчивается: найдены условный оптимальный выигрыш и условные оптимальное управление для каждого шага.

2.3.2. Окончательная (безусловная) оптимизация

Предположим, что исходное состояние S_0 нам полностью известно. Подставим это состояние S_0 в формулу для условного оптимального выигрыша $W_1^*(S_0)$.

Получим

$$W_{\max} = W_1^*(S_0),$$

а оптимальное управление на этом шаге

$$U_1 = u_1(S_0)$$

Далее, зная исходное состояние S_0^* и управление U_1 , можем найти состояние S_1^* системы после первого шага

$$S_1^* = \varphi_1(S_0^*, u_1).$$

Зная состояние S_1^* , можно найти оптимальное управление на втором шаге

$$U_2 = u_2(S_1^*),$$

а затем

$$S_2^* = \varphi_2(S_1^*, u_2)$$

и т.д.

Таким образом, идя по цепочке (рис. 2.6.), мы определим одним за другим все шаги оптимального управления и оптимальное управление операцией в целом

$$u = f(u_1, u_2, \dots, u_m).$$

На этом процесс оптимизации заканчивается.

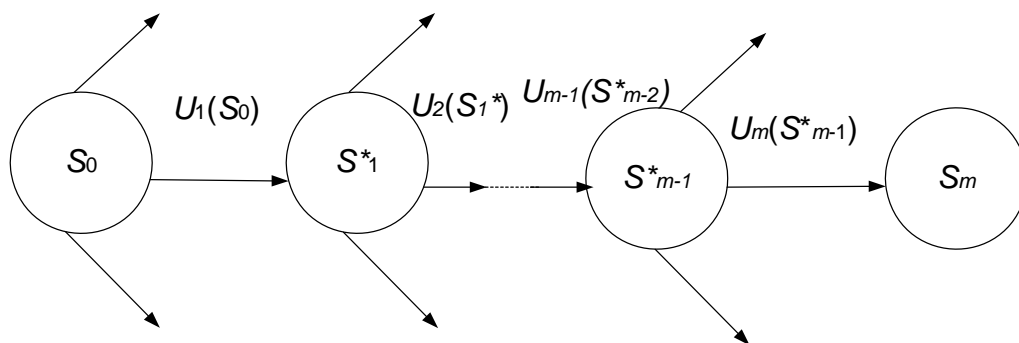


Рис. 2.6

Принципы и методы динамического программирования рассмотрим на примере задачи выбора кратчайшего маршрута на транспортной сети.

2.4. Задачи о маршрутизации

2.4.1. Постановка задачи

Найти кратчайший путь из пункта A в пункт B на сети, изображенной на рис. 2.7, где пункты обозначены кружками, а соединяющие их дороги отрезками (стрелками). Расстояния a_{ij} между пунктами проставлены над стрелками.

С точки зрения интересов оптимизации после каждого ближайшего шага (выбора кратчайшего расстояния из точки A_i в точку A_j). То следует двигаться по маршруту $A_0 - A_1 - A_3 - A_2 - A_4 - B$.

Длина этого маршрута ($4 + 7 + 5 + 10 + 8$) равна 34.

Решив эту задачу методом динамического программирования мы убедимся, что этот маршрут не является оптимальным.

A_1

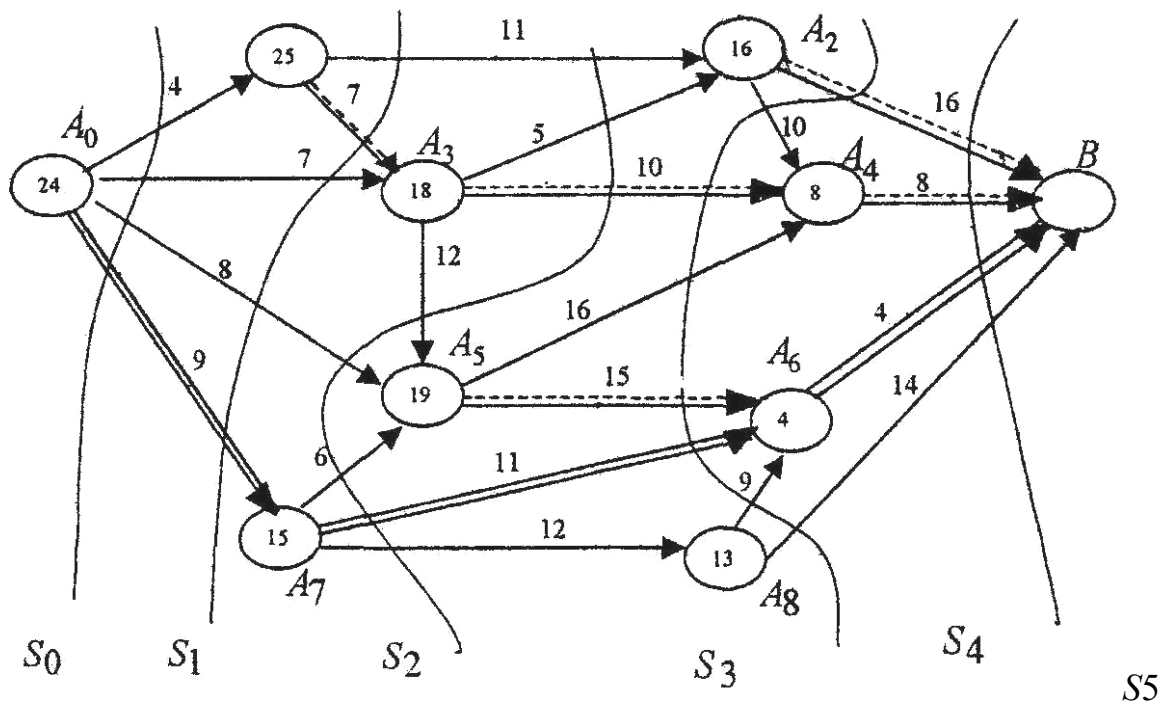


Рис. 2.7

2.4.2. Построение математической модели

Пусть задана ориентированная сеть, содержащая N точек (узлов). Найти кратчайший путь из точки 1 в точку N (рис. 2.8), если задана матрица a_{ij} расстояний из точки i в точку j .

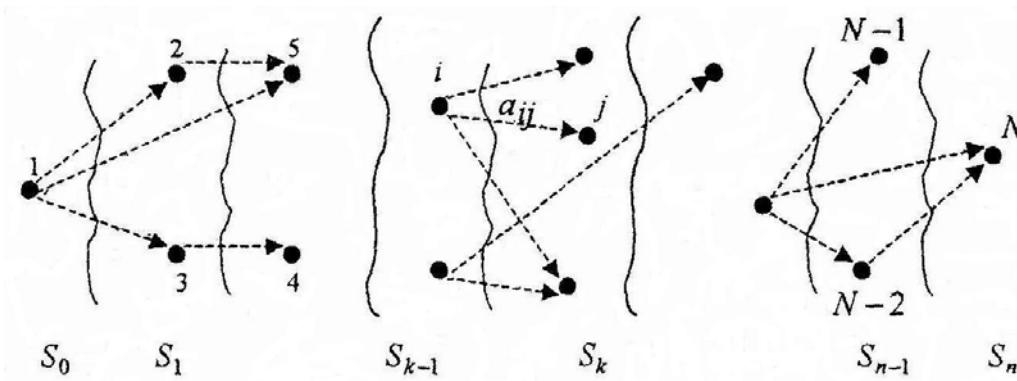


Рис. 2.8

Обозначим через W_j^* минимальный путь из точки i в точку N . Оптимальный маршрут из любой точки i должен обладать тем свойством, что каков бы ни был способ достижения пункта i последующее решение должно быть оптимальным для части пути, начинающегося в точке i (принцип оптимальности).

Пусть из точки i можем перейти в точку j , расстояние между этими точками равно a_{ij} . Точка j должна выбираться таким образом, чтобы путь из j в

N был частью оптимального из i в N . Обозначим минимальный путь из j в N через W_j^* . Тогда i выбирается из условия минимизации суммы

$$a_{ij} + W_j^*.$$

Таким образом получаем уравнение Беллмана.

$$W_i^* = \min_{j \neq i} \{a_{ij} + W_j^*\} \quad (2.3)$$

Для реализации уравнения (2.3) разделим условно все точки сети на n множеств по числу шагов 1, 2, ..., n (см. рис. 2.8.). К множеству S_0 отнесем точки, из которых можно попасть в N не более чем за n шагов, к S_i - точки из которых можно попасть в N не более чем за $n - 1$ шагов и т.д.

Если $i \in S_{k-1}$, то будем считать, что $j \in S_k$. Тогда уравнение, (2.3) примет вид

$$W_k^*(i) = \min_{\substack{i \in S_{k-1} \\ j \in S_k}} \{a_{ij} + W_{k+1}^*(j)\}.$$

Так как точка N единственна и относится к множеству S_n , тогда

$$W_{n+1}^*(N) = 0.$$

Множество S_{n-1} состоит из точек i , из которых можно попасть в N не более чем за один шаг, поэтому

$$W_n^*(i) = \min_{i \in S_{n-1}} \{a_{in}\} = a_{in}, \quad U_n^*(i) = N,$$

где $U_n^*(i)$ - условные оптимальное управление (решение) на n -ом переходе из точки i в N по кратчайшему пути.

Аналогично для точек $i \in S_{n-2}$

$$W_{n-1}^*(i) = \min_{\substack{i \in S_{n-2} \\ j \in S_{n-1}}} \{a_{ij} + W_n^*(j)\} = \min_{\substack{i \in S_{n-2} \\ j \in S_{n-1}}} \{a_{ij} + a_{jN}\}, \quad U_{n-1}^*(i)$$

и т.д. В итоге условной оптимизации получим совокупность условных оптимальных решений $U_k^*(i)$, используя которые последовательно определим точки, соответствующие оптимальному маршруту.

2.4.3. Последовательность решения задачи

Решение. Отнесем к множеству S_4 точки A_4 и A_6 , из которых можно попасть в точку B не более чем за один шаг; к S_3 - точки A_2 , A_5 и A_8 , из которых можно попасть в точку B не более чем за два шага; к S_2 - точки A_3 и A_7 , из которых можно попасть в точку B не более чем за три шага; к S_1 - точку A_1 , (не более чем за четыре шага до точки B), к S_0 - точку A_0 .

Условные оптимальные маршруты, начинающиеся в точке A_i и идущие в точку A_j , будем изображать дополнительной пунктирной стрелкой, а условные минимальные пути от A_i до B записывать в кружках точки A_i . Сначала найдем

$$W_5^*(A_4) = 8, \quad U_5^*(A_4) = B.$$

$$W_5^*(A_6) = 4, \quad U_5^*(A_6) = B.$$

Далее определим

$$W_4^*(A_2) = \min \begin{cases} 16 & \text{при } U_4(A_2) = B \\ 10 + 8 = 18 & \text{при } U_4(A_2) = A_4 \end{cases}$$

$$\text{т.е. } W_4^*(A_2) = 16, \quad U_4^*(A_2) = B.$$

$$W_4^*(A_5) = \min \begin{cases} 16 + 8 = 24 & \text{при } U_4(A_5) = A_4, \\ 15 + 4 = 19 & \text{при } U_4(A_5) = A_6, \end{cases}$$

$$\text{т.е. } W_4^*(A_5) = 19, \quad U_4^*(A_5) = A_6.$$

$$W_4^*(A_8) = \min \begin{cases} 9 + 4 = 13 & \text{при } U_4(A_8) = A_6, \\ 14 & \text{при } U_4(A_8) = B, \end{cases}$$

$$\text{т.е. } W_4^*(A_8) = 13, \quad U_4^*(A_8) = A_6.$$

$$W_3^*(A_3) = \min \begin{cases} 5 + 16 = 21 & \text{при } U_3(A_3) = A_2, \\ 10 + 8 = 18 & \text{при } U_3(A_3) = A_4, \\ 12 + 9 = 21 & \text{при } U_3(A_3) = A_5, \end{cases}$$

$$\text{т.е. } W_3^*(A_3) = 18, \quad U_3^*(A_3) = A_4.$$

$$W_3^*(A_7) = \min \begin{cases} 6 + 19 = 25 & \text{при } U_3(A_7) = A_5, \\ 11 + 4 = 15 & \text{при } U_3(A_7) = A_6, \\ 12 + 18 = 30 & \text{при } U_3(A_7) = A_8, \end{cases}$$

$$\text{т.е. } W_3^*(A_7) = 15, \quad U_3^*(A_7) = A_6.$$

$$W_2^*(A_1) = \begin{cases} 11 + 16 = 27 & \text{при } U_2(A_1) = A_2, \\ 7 + 18 = 25 & \text{при } U_2(A_1) = A_3, \end{cases}$$

$$\text{т.е. } W_2^*(A_1) = 25, \quad U_2^*(A_1) = A_3.$$

$$W_1^*(A_0) = \begin{cases} 4 + 25 = 29 & \text{при } U_1(A_0) = A_1, \\ 7 + 18 = 25 & \text{при } U_1(A_0) = A_3, \\ 8 + 19 = 27 & \text{при } U_1(A_0) = A_5, \\ 9 + 15 = 24 & \text{при } U_1(A_0) = A_7, \end{cases}$$

$$\text{т.е. } W_1^*(A_0) = 24, \quad U_1^*(A_0) = A_7.$$

Получаем, что минимальный путь равен $W_1^*(A_0) = 24$. Соответствующий маршрут проходит через точки A_0, A_7, A_6, B . На рис. 2.7 он выделен двойной линией.

2.5. Задачи замены оборудования

2.5.1. Постановка задачи

Одной из проблем, с которой приходится сталкиваться при организации работы автомобильного транспорта, является замена старого оборудования (станков, агрегатов, машин) и автомобилей на новые.

Старое оборудование (автомобиль) имеет физический и моральный износ, в результате чего растут производственные затраты по выпуску продукции на старом оборудовании, увеличиваются затраты на его ремонт и обслуживание, а вместе с тем снижаются его производительность и ликвидная стоимость.

Наступает момент, когда старое оборудование (автомобиль) более выгодно продать (заменить новым), чем эксплуатировать ценой больших затрат.

Оптимальная стратегия замены оборудования состоит в определении оптимальных сроков замены. Критериями оптимальности при определении сроков замены могут служить либо прибыль от эксплуатации, которую следует максимизировать, либо суммарные затраты на эксплуатацию в течение рассматриваемого промежутка времени, которые подлежат минимизации.

Условимся считать, что решение о замене оборудования принимается периодически в начале каждого промежутка (года), на который разбит плановый период.

Основными функциональными характеристиками оборудования являются:

t - возраст оборудования ($t = 0, 1, 2, 3, \dots, n$),

где $t = 0$ - использование нового оборудования,

$t = 1$ - использование оборудования возраста одного года и т.д.;

$f(t)$ - стоимость продукции (для автомобиля - выручка за транспортные услуги), произведенной за год на оборудовании возраста t ;

$r(t)$ - эксплуатационные затраты за год на оборудование возраста t ;

$\varphi(t)$ - остаточная стоимость оборудования возраста t ;

P - цена нового оборудования;

t_0 - начальный возраст оборудования;

n - продолжительность планового периода (количество лет в плановом периоде).

Схема возможных состояний оборудования может выглядеть так (рис. 2.9).

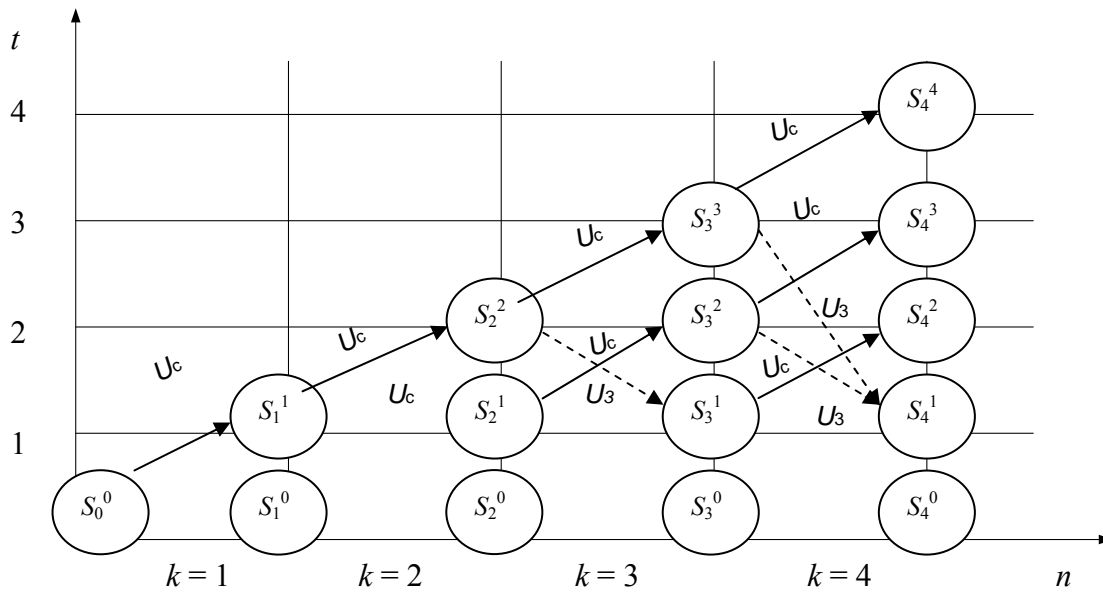


Рис. 2.9

где U^c - сохранность и продолжительность использования оборудования;
 U^3 - заменяемость оборудования новым;
 S_k^t - состояние оборудования, соответствующее возрасту t .

Для определения условных оптимальных решений необходимо составить функциональное уравнение Беллмана.

2.5.2. Построение математической модели

Поставленную задачу можно рассматривать как задачу динамического программирования, в которой в качестве системы S выступает оборудование. Состояния этой системы определяется фактическим временем использования оборудования (его возраста) t , т.е. описывается единым параметром t .

Алгоритм решения задачи методом ДП реализуется в два этапа.

Первый этап. При движении от начала n -го года к началу 1-го года для каждого допустимого состояния оборудования находится условное оптимальное управление (решение) - $u(t)$.

Второй этап. При движении от начала 1-го года к началу n -го года из условных оптимальных решений составляется оптимальный план замены оборудования - $u^*(t)$.

Рассмотрим n -шаговый процесс (см. рис. 2.9), считая k -м шагом номер k -го года от начала эксплуатации ($k = 1, 2, 3, \dots, n$). Уравнение на k -м шаге выбирается из двух возможных решений: u^c - сохранить и продолжить использование старого оборудования или u^3 - заменить оборудование новым.

Состояние S_{k-1} системы в начале k -го шага характеризуется параметром t - возраст оборудования, который может принимать значения $0, 1, 2, \dots, k - 1$, т.е. $t \leq k - 1$.

Если к началу k -го шага система находится в состоянии S_{k-1} и возраст ее равен t годам ($S_{k-1} = t$), то под влиянием уравнения U^c в конце k -го шага она перейдет в состояние S_k с возрастом оборудования $t + 1$ ($S_k = t + 1$) (рис. 7.9), т.е. возраст оборудования увеличится на один год. Под влиянием уравнения U^3 , принятого на k -м шаге, система перейдет в состояние с возрастом оборудования, равным одному году. Замену произвели в начале k -го шага ($S_k = 1$).

Определим прибыль на k -м шаге (показатель эффективности k -го шага) соответствующую каждому из альтернативных управлений U^c и U^3 .

Выбирая на k -м шаге управление U^c , мы сможем произвести продукцию стоимостью $f(t)$ на старом оборудовании, что потребует затрат $r(t)$, поэтому прибыль равна $f(t) - r(t)$. Обозначим ее через

$$W_k^c = f(t) - r(t).$$

При управлении U^3 получим доход $\varphi(t)$ от продажи старого оборудования (ликвидную стоимость) и $f(0)$ от произведенной на новом оборудовании продукции, затратив P рублей на приобретение нового оборудования, и $r(0)$ - на содержание нового оборудования. В этом случае прибыль составит

$$W_k^3 = \varphi(t) + f(0) - P - r(0).$$

Так как на последнем этапе процесса планирования мы можем действовать без учета предыдущих этапов и считать, что оптимальное управление на последнем этапе должно обеспечить максимальный доход за последний год, то функциональное уравнение, отражающее возможные решения, будет следующим:

$$W_n^*(t) = \max \begin{cases} f(t) - r(t) & \text{при } U_n = U^c, \\ \varphi(t) + f(0) - P - r(0) & \text{при } U_n = U^3 \end{cases} \quad (2.4)$$

Сравнив эти две величины для всех возможных $i < n$ получим значение $W_n^*(t)$ и соответствующее значение оптимального управления $U_n^*(t)$.

Предположим, что для всех значений t S_k - о состоянии системы известна максимальная прибыль, полученная за $n - k$ шагов с $k + 1$ -го по n -й включительно. Поэтому основные рекуррентные соотношения можно записать в виде

$$W_k^*(t) = \max \begin{cases} f(t) - r(t) + W_{k+1}^*(t+1) & \text{при } U_k = U^c, \\ \varphi(t) + f(0) - P - r(0) + W_{k+1}^*(1) & \text{при } U_k = U^3. \end{cases} \quad (2.5)$$

В уравнении (2.5) величина $W_{k+1}^*(1)$ - условная максимальная прибыль, полученная за $n - k$ шагов, если к началу $(k + 1)$ -го шага системы находились в состоянии S_k и $t = 1$ (возраст оборудования составлял 1 года).

2.5.3. Последовательность решения задачи

Решение задачи замены оборудования методом динамического программирования рассмотрим на конкретном примере.

Пример. Составить план замены оборудования в течение пяти лет, при котором общая прибыль за данный период времени максимальна, если затраты, связанные с приобретением и установкой нового оборудования составляют 40 тыс. руб. Зависимость производительности оборудования от времени его использования, а также зависимость затрат на содержание и ремонт оборудования при различном времени его использования приведены в табл. 2.1.

Таблица 2.1

Время, в течение которого используется оборудование, год	0	1	2	3	4	5
Годовой выпуск продукции $f(t)$ в стоимостном выражении, тыс. руб.	80	75	65	60	60	55
Эксплуатационные затраты, $r(t)$ тыс. руб.	20	25	30	35	45	55

1. Нахождение решения исходной задачи начинаем с определения условного оптимального управления (решения) для последнего 5-го года и находим множество допустимых состояний оборудования к началу данного года. Так как в начальный момент имеется новое оборудование ($t^{(1)} = 0$), то возраст оборудования к началу 5-го года может составлять 1, 2, 3, 4 года. Поэтому допустимые состояния системы на данный период времени таковы: $t_1^{(5)} = 1$; $t_2^{(5)} = 2$, $t_3^{(5)} = 3$, $t_4^{(5)} = 4$. Для каждого из этих состояний найдем условное оптимальное решение и соответствующее значение функции $W_5(t^{(5)})$.

Используя уравнение (2.4) и соотношение $W_6(t^{(k+1)}) = 0$ (так как рассматривается последний год расчетного периода), получаем:

$$W_5(t^{(5)}) = \max \left\{ \begin{array}{l} f(t^{(5)}) - r(t^{(5)}) \\ f(t^{(5)} = 0) - r(t^{(5)} = 0) - p \end{array} \right\}. \quad (2.6)$$

Подставляя теперь в формулу (2.6) вместо $t^{(5)}$ его значение, равное 1; и учитывая данные таблицы 2.1, находим:

$$W_5(t_1^{(5)}) = \max \left\{ \begin{array}{l} f(t^{(5)} = 1) - r(t^{(5)} = 1) \\ f(t^{(5)} = 0) - r(t^{(5)} = 0) - P \end{array} \right\} = \max \left\{ \begin{array}{l} 75 - 25 \\ 80 - 20 - 40 \end{array} \right\} = 50, \quad U = U^c.$$

Значит, условное оптимальное решение в данном случае - сохранить оборудование.

Проведем аналогичные вычисления для других допустимых состояний оборудования к началу 5 - го года.

$$W_5(t_2^{(5)}) = \max \left\{ \begin{array}{l} 65 - 30 \\ 80 - 20 - 40 \end{array} \right\} = 35, \quad U = U^c,$$

$$W_5(t_3^{(5)}) = \max \left\{ \begin{array}{l} 65 - 35 \\ 80 - 20 - 40 \end{array} \right\} = 25, \quad U = U^c,$$

$$W_5(t_4^{(5)}) = \max \left\{ \begin{array}{l} 60 - 45 \\ 80 - 20 - 40 \end{array} \right\} = 20, \quad U = U^3.$$

Полученные результаты сводим в табл. 2.2.

Таблица 2.2

Возраст оборудования $t^{(5)}$, год	Значение функции $W_5(t^{(5)})$, тыс. руб.	Условное оптимальное решение, U
1	50	U^c
2	5	U^c
3	25	U^c
4	20	U^3

2. Рассмотрим теперь возможные состояния оборудования к началу 4-го года. Очевидно, что допустимыми состояниями являются $t_1^{(4)} = 1$, $t_2^{(4)} = 2$, $t_3^{(4)} = 3$. Для каждого из них определяется условное оптимальное решение и соответствующее значение функции $W_4(t^{(4)})$. Для этого используем уравнение (2.5) и данные табл. 2.1, 2.2. Так, в частности, для $t_1^{(4)}$ имеем

$$W_4(t_1^{(4)}) = \max \left\{ \begin{array}{l} f(t^{(4)} = 1) - r(t^{(4)} = 1) + W_5(t^{(5)} = 2) \\ f(t^{(4)} = 0) - r(t^{(4)} = 0) - P + W_5(t^{(5)} = 1) \end{array} \right\} =$$

$$= \max \left\{ \begin{array}{l} 75 - 25 + 35 \\ 80 - 20 - 40 + 50 \end{array} \right\} = 85, \quad U = U^c.$$

Значит, условное оптимальное решение в данном случае - сохранить оборудование.

Проведем аналогичные вычисления для других допустимых состояний оборудования к началу 4-го года.

$$W_4(t_2^{(4)}) = \max \left\{ \begin{array}{l} 65 - 30 + 25 \\ 80 - 20 - 40 + 50 \end{array} \right\} = 70, \quad U = U^3,$$

$$W_4(t_3^{(4)}) = \max \left\{ \begin{array}{l} 65 - 35 + 20 \\ 80 - 20 - 40 + 50 \end{array} \right\} = 70, \quad U = U^3.$$

Полученные результаты вычислений занесем в табл. 2.3.

Таблица 2.3

Возраст оборудования $t^{(4)}$ год	Значение функции $W_4(t^{(4)})$, тыс. руб.	Условное оптимальное решение, U
1	85	U^c
2	70	U^c
3	70	U^3

3. Определяем теперь условное оптимальное решение для каждого из допустимых состояний оборудования к началу 3-го года. Очевидно, такими состояниями являются $t_1^{(3)} = 1$, $t_2^{(3)} = 2$. В соответствии с уравнением (2.5) и табл. 2.1, 2.3 имеем

$$\begin{aligned}
 W_3(t_1^{(3)}) &= \max \left\{ \begin{array}{l} f(t^{(3)} = 1) - r(t^{(3)} = 1) + W_4(t^{(4)} = 2) \\ f(t^{(3)} = 0) - r(t^{(4)} = 0) - P + W_4(t^{(4)} = 1) \end{array} \right\} = \\
 &= \max \left\{ \begin{array}{l} 75 - 25 + 70 \\ 80 - 20 - 40 + 85 \end{array} \right\} = 120, \quad U = U^c \\
 W_3(t_2^{(3)}) &= \max \left\{ \begin{array}{l} f(t^{(3)} = 2) - r(t^{(3)} = 2) + W_4(t^{(4)} = 3) \\ f(t^{(3)} = 0) - r(t^{(4)} = 0) - P + W_4(t^{(4)} = 1) \end{array} \right\} = \\
 &= \max \left\{ \begin{array}{l} 65 - 30 + 70 \\ 80 - 20 - 40 + 85 \end{array} \right\} = 105, \quad U = U^c (U^3)
 \end{aligned}$$

Из последнего выражения видно, что если к началу 3-го года возраст оборудования составляет 2 года, то независимо от того, будет ли принято решение U^c или U^3 , величина прибыли окажется одной и той же. Это означает, что в качестве условного оптимального решения можно взять любое, например U^c . Полученные значения для $W_3(t_1^{(3)})$ и соответствующие условные оптимальные записываем в табл. 2.4.

Таблица 2.4

Возраст оборудования $t^{(3)}$ год	Значение функции $W_3(t^{(3)})$, тыс. руб.	Условное оптимальное решение, U
1	120	U^c
2	105	U^c

1. Наконец, рассмотрим допустимые состояния оборудования к началу 2-го года. Очевидно, что на данный момент времени возраст оборудования может быть равен только одному году. В соответствии с уравнением (2.5) имеем

$$W_2(t_1^{(2)}) = \max \left\{ \begin{array}{l} f(t^{(2)} = 1) - r(t^{(2)} = 1) + W_3(t^{(3)} = 2) \\ f(t^{(2)} = 0) - r(t^{(2)} = 0) - P + W_3(t^{(3)} = 1) \end{array} \right\} =$$

$$= \max \left\{ \begin{array}{l} 75 - 25 + 105 \\ 80 - 20 - 40 + 120 \end{array} \right\} = 155, \quad U = U^c.$$

Результат занесем в табл. 2.5.

Таблица 2.5

Возраст оборудования $t^{(2)}$ год	Значение функции $W_2(t^{(2)})$, тыс. руб.	Условное оптимальное решение, U
1	155	U^c

5. Согласно условию в начальный момент установлено новое оборудование ($t_1^{(1)} = 0$). Поэтому проблемы выбора между сохранением и заменой оборудования не существует оборудование следует сохранить. Значит, условным оптимальным решением является U^c , и значение функции $W_1(t_1^{(1)}) = f(t_1^{(1)} = 0) - r(t_1^{(1)} = 0) + W_2(t_1^{(1)} = 1) = 80 - 20 + 155 = 215$.

6. В результате реализации второго этапа вычислительного процесса, состоящего в прохождении всех рассмотренных шагов с начала первого до начала пятого года, максимальная прибыль предприятия может быть равна 215 тыс. руб., что соответствует оптимальному плану замены оборудования.

7. Для 1-го года решение единственное - надо сохранить оборудование. Значит, возраст оборудования к началу 2-го года равен одному году, и оборудование (согласно табл. 2.5) надо сохранить. Реализация такого решения приводит к тому, что возраст оборудования к началу 3-го года становится равным двум годам, и оборудование (согласно табл. 2.4) надо сохранить, т.е. его возраст к началу 4-го года становится равным трем годам, и оборудование (согласно табл. 2.3) надо заменять. После замены оборудования его возраст к началу 5-го года составит один год. Как видно из табл. 7.2, при таком возрасте оборудования его менять не следует. Итак, получается следующий оптимальный план замены оборудования (рис. 2.10).

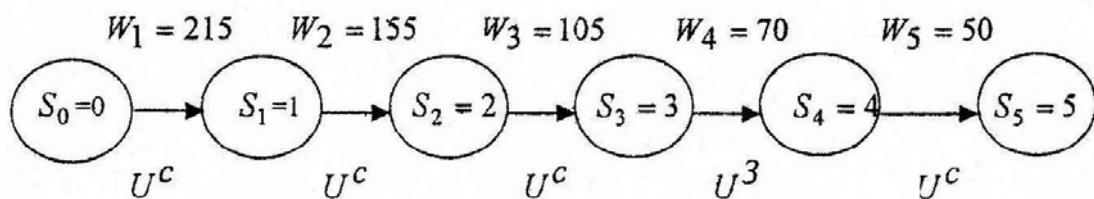


Рис. 2.10

Контрольные вопросы

1. Какие задачи автомобильного транспорта решаются методами динамического программирования?
2. Сформулируйте общую задачу динамического программирования.
3. Перечислите принципы оптимизации задач динамического программирования.
4. Запишите основные уравнения динамического программирования (уравнение Беллмана) и перепишите его составляющие.
5. Особенности предварительной (условной) оптимизации.
6. Особенности окончательной (безусловной) оптимизации.
7. Сформулируйте задачу о маршрутизации.
8. Запишите математическую модель решения задачи о маршрутизации методом динамического программирования.
9. Последовательность решения задачи о маршрутизации методом динамического программирования.
10. Сформулируйте задачу о замене оборудования.
11. Запишите математическую модель решения задачи замены оборудования методом динамического программирования.
12. Последовательность решения задачи замены оборудования методом динамического программирования.

2) Матричная форма записи:

$$W = CX.$$

При ограничениях

$$AX \leq B,$$

где $C = (C_1, C_2, \dots, C_n)$ – матрица-строка;

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} - \text{матрица-столбец};$$

$$A = (a_{ij}) - \text{матрица системы} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix};$$

CX и AX - произведение матриц;

$$B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} - \text{матрица-столбец.}$$

3) Запись с помощью знаков суммирования

$$W = \sum_{j=1}^n C_j x_j \rightarrow (\max (\min)).$$

При ограничениях

$$\sum_{j=1}^n a_{ij} x_{ij} \leq b_i; \quad i = 1, 2, \dots, m.$$

Система уравнений (8.2) определяет область решения ЗЛП.

Изобразим область решений ЗЛП для двумерного случая, т.е. (x_1 и x_2) на плоскости (см. рис. 8.1.), тогда:

- 1) B1, B2, B3, B4, B5, B1 - область решения ЗЛП.
- 2) A1, B2, B3, A2, 0, Ф1 - область допустимых решений, т.к. $X_j \geq 0$.
- 3) Вершины многоугольника, для которых $X_{1,2} \geq 0$. 0, A1, B2, B3, A2 называются опорными точками.
- 4) Значения целевой функции в опорных точках называют опорными решениями.
- 5) В одной из вершин 0, A2, B2, B3, A2 $W \rightarrow \max (\min)$, которые называют оптимальным решением, т.е. только в одной точке можем иметь оптимальное решение.

где $Z(y) = b_1y_1 + b_2y_2 + \dots + b_m y_m \rightarrow \max$.

Примечание.

1. Двойственная задача к двойственной - есть прямая задача.
2. Переход к двойственной задаче позволяет сократить объемы вычислений.
3. По решению одной из двойственных задач находится оптимальное решение другой, что подтверждается теорией двойственности.

Теорема (двойственности)

Если из пары двойственных задач одна обладает оптимальным решением, то и другая имеет решение, причем для экстремальных значений целевой функции выполняется соотношение

$$\min W(x) = \max z(y).$$

Пример. Пусть имеем исходную задачу линейного программирования

$$W = 2x_1 + x_2 + 5x_3 \rightarrow \min$$

при ограничениях

$$\left. \begin{array}{l} -y_1 + y_2 + 2y_3 \leq 2 \\ y_1 - 5y_2 - y_3 \leq 1 \\ y_1 + y_2 + 3y_3 \leq 5 \end{array} \right\} x_j \geq 0 (j = 1, 2, 3)$$

Требуется записать двойственную задачу.

Решение.

$$Z = -4y_1 + 5y_2 + 6y_3 \rightarrow \max$$

при ограничениях

$$\left. \begin{array}{l} -y_1 + y_2 + 2y_3 \leq 2 \\ y_1 - 5y_2 - y_3 \leq 1 \\ y_1 + y_2 + 3y_3 \leq 5 \end{array} \right\} y_i \geq 0 (i = 1, 2, 3).$$

3.3. Формулировка задачи линейного программирования

В общем виде задачу можно сформулировать так:

Предприятие может изготовить изделия двух видов A_1 и A_2 . Для изготовления этих изделий необходимо иметь три вида сырья B_1 , B_2 и B_3 , ресурсы которых ограничены. На производство одного изделия требуется определенное количество сырья, для конкретного случая они представлены табл. 3.1. В этой таблице также приведены доходы от реализации от одного изделия в условных единицах.

Требуется определить, сколько и каких видов изделий надо производить, чтобы доход предприятия был максимальным.

Таблица 3.1

Вид сырья	Вид продукции		Запас сырья (B_i)
	A_1	A_2	
B_1	$a_{11} = 1$	$a_{12} = 4$	28
B_2	$a_{21} = 1$	$a_{22} = 1$	10
B_3	$a_{31} = 3$	$a_{32} = 1$	24
Доход от одного вида продукции	$C_1 = 3$	$C_2 = 2$	
Количество единиц продукции	X_1	X_2	

Используя данные табл. 3.1, запишем целевую функцию нашей задачи.

$$\max W = C_1 x_1 + C_2 x_2 = 3x_1 + 2x_2, \quad (3.4)$$

где C_1, C_2 - стоимость изделий;

x_1, x_2 - количество изделий.

Система ограничений запишется так:

$$\left. \begin{aligned} a_{11} + a_{12}x_2 &\leq B_1 \\ a_{21} + a_{22}x_2 &\leq B_2 \\ a_{31} + a_{32}x_2 &\leq B_3 \end{aligned} \right\}$$

или

$$\left. \begin{aligned} x_1 + 4x_2 &\leq 28 \\ x_1 + x_2 &\leq 10 \\ 3x_1 + x_2 &\leq 21 \end{aligned} \right\} \text{ при } x_{1,2,3} \geq 0. \quad (3.5)$$

Итак, задача линейного программирования (ЗЛП) сформулирована в словесной и математической форме.

ЗЛП имеет несколько методов решения. Мы рассмотрим графический метод и метод симплексных таблиц.

3.4. Геометрическая интерпретация задачи линейного программирования

Поскольку в задаче только две переменные, а условия для них записаны в виде неравенств первой степени, можно рассмотреть геометрическую интерпретацию задачи в плоскости координат x_1, x_2 .

Условия (3.5), накладываемые на переменные x_1, x_2 означают, что область допустимых значений x_1, x_2 для каждого условия лежит в заштрихованной полуплоскости (рис. 3.2).

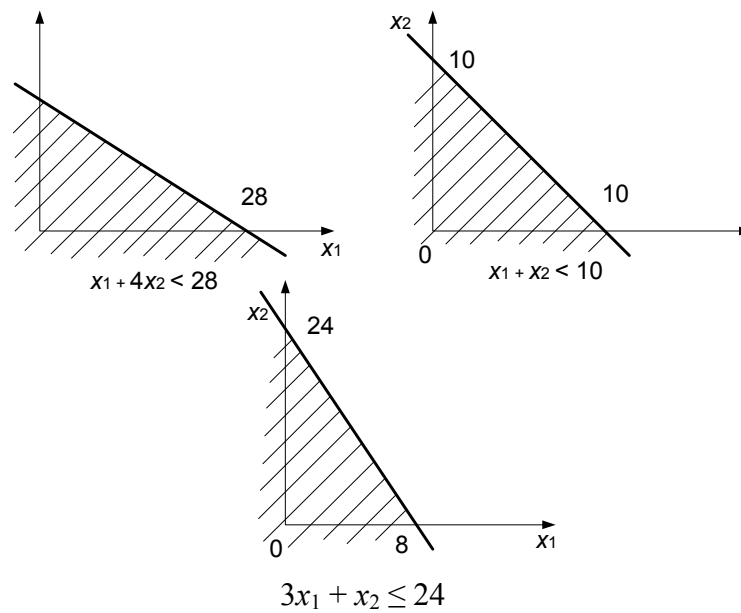


Рис. 3.2

Область изменения x_1, x_2 , удовлетворяющая всем трем условиям, лежит в выпуклом многоугольнике $0A_1 A_2 A_3 A_4$ с вершинами $0(0; 0)$, $A_1(0; 7)$, $A_2(6; 4)$, $A_3(7; 3)$, $A_4(8; 0)$ (рис. 3.3).

Рассмотрим теперь линии уровня функции $W(x_1, x_2)$, семейство параллельных прямых $3x_1 + 2x_2 = C$.

Для $W(x_1, x_2) = 0$ линия пройдет через начало координат. Для больших C линии уровня расположены правее. Следовательно, максимум функции $W(x_1, x_2) = 3x_1 + 2x_2$ реализуется в точке (x_1, x_2) , лежащей на линии уровня функции $W(x_1, x_2) = C$, которая расположена на наибольшем расстоянии от нуля вправо. С другой стороны, мы ищем наибольшее значение $W(x_1, x_2)$ в области изменения переменных $0 A_1 A_2 A_3 A_4$. Следовательно, оптимальное решение лежит на линии уровня $3x_1 + 2x_2 = C$, пересекающей область $0 A_1 A_2 A_3 A_4$ в точке A_3 . Таким образом, точка $A_3(7; 3)$ является точкой оптимального решения, и

$$\max W(x_1, x_2) = 3 \cdot 7 + 2 \cdot 3 = 27.$$

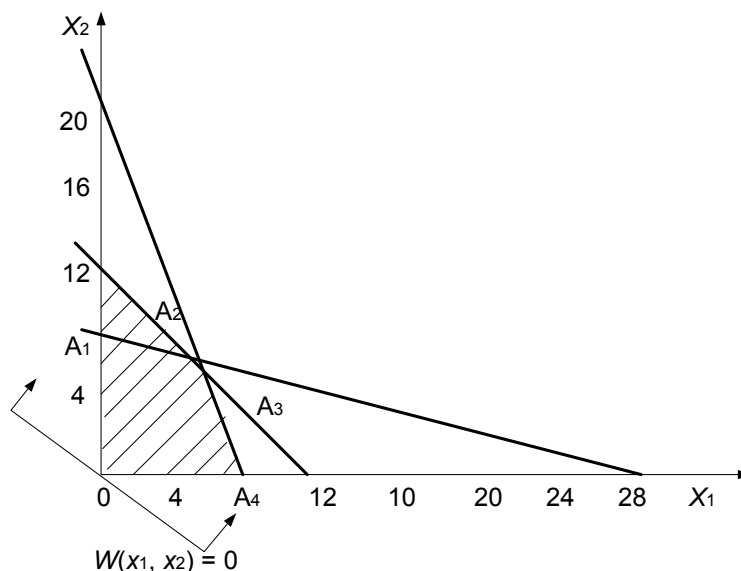


Рис. 3.3

3.5. Симплексный метод решения задач линейного программирования

Пусть требуется найти \max (\min) целевой функции

$$W = \sum_{j=1}^n C_j x_j$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \leq b_j ; \quad i = 1, 2, \dots, m,$$

где $x_j \geq 0$; a_{ij} , b_i , C_j - постоянные величины (коэффициенты), т.е. имеем ЗЛП.

Для решения ЗЛП применяют различные числовые методы. Одним из наиболее распространенных (универсальных) методов является симплексный метод (табличный).

Для решения ЗЛП симплекс-методом предварительно необходимо провести следующие преобразования:

1) Уравнения системы ограничений (3.2) записать в каноническом виде, для чего заменяем неравенства (3.2) равенствами путем введения дополнительных неотрицательных переменных. Получаем:

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} &= b_2, \\ \dots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} &= b_m, \end{aligned} \right\} \quad (3.6)$$

где x_{n+1} , x_{n+2} , ..., x_{n+m} - некоторые новые неотрицательные переменные, которые называют добавочными (фиктивными).

2) Дополнительные переменные можно ввести и в целевую функцию с нулевыми коэффициентами. Тогда получим

$$W(x) = c_1x_1 + c_2x_2 + \dots + c_nx_n + 0x_{n+1} + 0x_{n+2} + \dots + 0x_{n+m} \quad (3.7)$$

Примечание:

1. запись ЗЛП в виде уравнений 8.6 и 8.7 называют канонической;
2. добавочные переменные x_{n+1} и т.д. положительны, как и x_1, \dots, x_n ;
3. общее количество переменных будет $n_1 = n + m$, где n - первоначальные, m - добавочные (равно числу уравнений);
4. всегда возможен обратный переход.

На первый взгляд переход ЗЛП к каноническому виду усложняет задачу, но оказывается, что это позволяет более эффективно провести исследование задачи на оптимальность.

3.5.1. Основные положения симплекс-метода

Приведем уравнения (3.4), (3.5) к канонической форме. Для этого введем дополнительные переменные x_3, x_4, x_5 и запишем задачу в виде

$$\left. \begin{aligned} x_1 + 4x_2 + x_3 &\leq 28, \\ x_1 + x_2 + x_4 &\leq 10, \\ 3x_1 + x_2 + x_5 &\leq 24, \end{aligned} \right\} \quad (3.8)$$

где $x_1 \geq 0; x_2 \geq 0; x_3 \geq 0; x_4 \geq 0; x_5 \geq 0$.

$$W = 3x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5. \quad (3.9)$$

Следует найти x_1, x_2, x_3, x_4, x_5 , которые удовлетворяют условиям (8.8) обеспечивают максимум целевой функции $W(x_1, x_2, x_3, x_4, x_5)$.

Поиск оптимального решения начнем с выбора значений, удовлетворяющих условиям (3.8). Легче всего это сделать для $x_1 = x_2 = 0$, тогда $x_3 = 28, x_4 = 10, x_5 = 24$. Такое начальное приближение оптимального решения назовем опорным решением: $x_1^{(1)} = 0, x_2^{(1)} = 0, x_3^{(1)} = 28, x_4^{(1)} = 10, x_5^{(1)} = 24$. Ему соответствует значение целевой функции $W^{(1)} = 0$.

Выразим из условия (3.8) ненулевые переменные x_3, x_4, x_5 (назовем их базисными переменными) через x_1, x_2 (назовем их свободными):

$$\begin{aligned} x_3 &= 28 - x_1 - 4x_2, \\ x_4 &= 10x_1 - x_2, \\ x_5 &= 24 - 3x_1 - x_2. \end{aligned}$$

Целевая функция в переменных x_1, x_2 такова: $W(x_1, x_2) = 3x_1 + 2x_2$. Видно, что увеличивая x_1, x_2 , можно увеличить значения целевой функции. Пусть, например, $x_2 = 0$. Тогда максимально допустимое значение $x_1 = 8$ (см. рис. 3.3). Получим новое опорное решение $x_1^{(2)} = 8, x_2^{(2)} = 0, x_3^{(2)} = 20, x_4^{(2)} = 2, x_5^{(2)} = 0$. Ему соответствует значение целевой функция $W^{(2)} = 24 > W^{(1)}$. Следовательно, второе опорное решение лучше.

Примем теперь нулевые переменные (x_2 и x_5) за свободные, а остальные (x_1, x_3, x_4) - за базисные. Из системы (3.8) получим

$$\begin{aligned}x_1 &= 8 - (1/3)x_2 - (1/3)x_5, \\x_3 &= 20 - (10/3)x_2 + (1/3)x_5, \\x_4 &= 2 - (2/3)x_2 + (1/3)x_5.\end{aligned}\tag{3.10}$$

В новых свободных переменных целевая функция имеет вид

$$W = 24 + x_2 - x_5.$$

Таким образом, для увеличения целевой функции W нужно увеличить значение свободной переменной x_2 , оставляя $x_5 = 0$. Из формулы (8.10) видно, что наибольшее допустимое значение $x_2 = 3$. Тогда новое опорное решение таково: $x_1^{(3)} = 7$, $x_2^{(3)} = 3$, $x_3^{(3)} = 10$, $x_4^{(3)} = 0$, $x_5^{(3)} = 0$, а значение целевой функции $W^{(3)} = 3 \cdot 7 + 2 \cdot 3 = 27 > W^{(2)}$.

Как и прежде за свободные переменные выберем x_4 и x_5 , а за базисные - x_1 , x_2 и x_3 . Из системы (3.8) получим

$$\begin{aligned}x_1 &= 7 + x_4/2 - x_5/2, \\x_2 &= 3 - 3x_4/2 + x_5/2, \\x_3 &= 9 - 11x_4/2 - 3x_5/2.\end{aligned}$$

Целевая функция имеет вид $W = 27 - (3/2)x_4 - (1/5)x_5$. Видно, что нельзя изменить значения x_4 и x_5 , не уменьшив значение целевой функции. Следовательно, последнее опорное решение оптимально. Как и при решении задачи геометрическим способом, получено оптимальное решение $x_1 = 7$, $x_2 = 3$, дающее максимальное значение функции

$$\begin{aligned}W(x_1, x_2) &= 3x_1 + 2x_2 \\x_1 &= 7, x_2 = 3.\end{aligned}$$

Описанный метод решения — симплекс - метод — состоит в построении последовательности опорных решений, увеличивающих целевую функцию.

Изобразим опорные решения в плоскости x_1, x_2 , помечая точки цифрой, соответствующей номеру опорного решения (рис. 3.4).

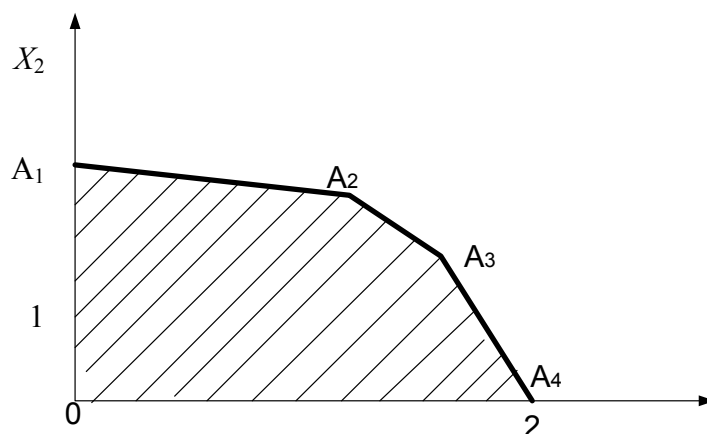


Рис. 3.4

Таким образом мы строим опорные решения симплекс-методом, переходя от начала координат к оптимальному решению по периметру многоугольника $O A_1 A_2 A_3 A_4$, который был получен при решении задачи геометрическим способом.

Из описания метода на примере видно, что алгоритм симплекс-метода носит итерационный характер. Один шаг метода состоит в построении очередного опорного решения.

3.5.2. Алгоритм симплекс-метода

Опишем алгоритм симплекс-метода для общей канонической задачи линейного программирования (3.6), (3.7). Допустимым решением назовем всякое решение системы (3.6), удовлетворяющее условию $x_i \geq 0$. Допустимое решение, на котором целевая функция (3.7) принимает максимальное значение, назовем оптимальным.

В зависимости от матрицы системы (3.6) возможны следующие случаи:

1. Система (3.6) не имеет решений, т.е. несовместима. В этом случае задачи линейного программирования не разрешимы.

2. Система (3.6) не имеет допустимых решений, а значит, задача линейного программирования не имеет решения.

3. Система (3.6) имеет единственное допустимое решение. В этом случае единственное решение системы (3.6) является оптимальным.

4. Система (3.6) имеет бесконечное множество допустимых решений, но целевая функция не ограничена на этом множестве допустимых решений, а, значит, задача линейного программирования не разрешима.

5. Система (3.6) имеет множество допустимых решений, и целевая функция ограничена на множестве допустимых решений. В этом случае задача линейного программирования имеет оптимальное решение. Будем в дальнейшем считать, что задача (3.6), (3.7) имеет оптимальное решение.

Симплекс-метод состоит в последовательном построении таких опорных решений, что каждое следующее увеличивает целевую функцию и максимум достигается за конечное число шагов.

Обобщенный алгоритм симплексного метода решения ЗЛП приведен на рис. 3.5 и включает следующую последовательность операций:

1. Уравнения системы ограничений запишем в каноническом виде. Для этого:

а) заменим неравенства равенствами путем введения дополнительных (фиктивных) переменных;

б) добиваемся, чтобы все свободные члены системы ограничений были положительными (блок 2).

2. Находим первое опорное решение. Для этого:
- а) все переменные n разбиваем на m -базисных и $k = n - m$ -свободных;
 - б) в качестве базисных переменных выбираем добавочные переменные;
 - в) полагая, что все свободные переменные равны нулю получим первое опорное решение (блок 3).



Рис. 3.5

3. По правилам 1, 2 (см. ниже) определяем оптимальность принятого решения (блоки 5 - 7).

4. При отсутствии оптимальности по определенному правилу (см. ниже) удаляем из базисных переменных одну и вводим другую из числа свободных (блоки 8, 9).

5. Проводим повторную проверку на оптимальность.

Примечание. Для упорядочения перехода от одного опорного решения пользуемся итерационными таблицами. Итерационные таблицы, составленные по нижеприведенным правилам, позволяют осуществлять переход от худшего решения только к лучшему, до получения оптимального решения.

Аналитический способ решения задачи линейного программирования рассмотрим на конкретном примере.

3.5.3. Решение задач линейного программирования

Задача. Найти максимум целевой функции

$$W = 3x_1 + 2x_2 \rightarrow \max$$

при ограничениях

$$\left. \begin{aligned} x_1 + 4x_2 &\leq 28, \\ x_1 + x_2 &\leq 10, \\ 3x_1 + x_2 &\leq 24, \end{aligned} \right\}$$

где $x_1, x_2 \geq 0$.

Решение.

1. Приведем систему ограничений к каноническому виду.

$$\begin{aligned} x_1 + 4x_2 + x_3 + 0 + 0 &= 28, \\ x_1 + x_2 + 0 + x_4 + 0 &= 10, \\ 3x_1 + x_2 + 0 + 0 + x_5 &= 24, \end{aligned}$$

где $x_3, x_4, x_5 \geq 0$ - добавочные переменные, которые, примем за базис.

Целевую функцию запишем в виде

$$W_{\max} = 3x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5.$$

1. Составляем первую симплексную таблицу (табл. 3.2).

Таблица 38.2

Базисные переменные		C_j		Коэффициент целевой функции					$Q_i = b_i/a_{ij}$
		C_i	b_i	x_1	x_2	x_3	x_4	x_5	
1	x_3	0	28	1	4	1	0	0	28
2	x_4	0	10	1	1	0	1	0	10
3	x_5	0	24	3	1	0	0	1	8(min)
$\Delta_j = \sum c_i a_{ij} - c_j$				-3	-2	0	0	0	
				$W = \sum c_i b_i = 0$					

где x_3, x_4, x_5 - базисные переменные;

C_j - коэффициенты целевой функции или неизвестные x_j ;

C_i - коэффициенты целевой функции при базисных переменных;

b_i - свободные члены системы ограничений;

a_{ij} - коэффициенты при неизвестной системе ограничений;

$\Delta_j = \sum c_i a_{ij} - c_j$ - оценочный параметр по j -му столбцу;

$Q_i = b_i/a_{ij}$ - оценочный параметр по i -й строке;

$W = \sum c_i b_i$ - значение целевой функции при принятых базисных переменных x_3, x_4, x_5 (свободные переменные x_1, x_2 равны нулю).

Правила проверки на оптимальность:

Правило 1.

Если при отыскании максимума целевой функции для некоторого опорного решения $\{x_i\}$ выполняется условие $\Delta_j = \sum c_i a_{ij} - c_j \geq 0$, то решение $\{x_i\}$ является оптимальным.

Для нашего примера $\Delta_1 = -3$ и $\Delta_2 = -2$, значит, план не оптимален.

Используя оценочный параметр Δ_j определим разрешающий столбец, т.е. из какого столбца и какую свободную переменную будем вводить в число базисных переменных.

Для нашего примера самый отрицательный оценочный параметр

$\Delta_1 = -3$ и, следовательно, переменную x_1 введем в число базисных переменных.

Правило 2.

Разрешающую (направляющую) - строку определяем с помощью оценочного параметра $Q_i = b_i/a_{ij}$, который должен быть положительным и минимальным для всех b_i/a_{ij} для разрешающего столбца.

Для нашего примера разрешающей строкой будет третья (см. табл. 8.2), т.к. $Q_3 = 24/3 = 8$ - самое минимальное положительное число. Таким образом, из базиса удаляем x_5 и вводим вместо него x_1 . Составляем новую итерационную (табл. 3.3).

Таблица 3.3

Базисные переменные		C_j		Коэффициент целевой функции					$Q_i = b_i/a_{ij}$
		C_i	b_i	x_1	x_2	x_3	x_4	x_5	
1	x_3	0	20	0	11/3	1	0	0	60/11
2	x_4	0	2	0	2/3	0	1	0	3(min)
3	x_1	3	8	1	1/3	0	0	0	24
$\Delta_j = \sum c_i a_{ij} - c_j$				0	-1	0	0	0	
				$W = \sum c_i b_i = 0 \cdot 20 + 0 \cdot 2 + 3 \cdot 8 = 24$					

Таблица 3.4

Базисные переменные		C_j		Коэффициент целевой функции					$Q_i = b_i/a_{ij}$
		C_i	b_i	x_1	x_2	x_3	x_4	x_5	
1	x_3	0	20	0	0	1	0	0	
2	x_2	0	2	0	1	0	0	0	
3	x_1	3	8	1	0	0	0	0	
$\Delta_j = \sum c_i a_{ij} - c_j$				0	0	0	0	0	
				$W = \sum c_i b_i = 3 \cdot 7 + 2 \cdot 3 = 27$					

3. При заполнении этой расчетной таблицы добиваемся, чтобы на пересечении разрешающей строки и разрешающего столбца была единица, а остальные элементы a_{1i} разрешающего столбца были равны нулю. Для этого:

- а) третью строку делим на 3;
- б) из первой и второй строки вычитаем третью строку (ранее поделенную на 3).

Для нашего примера $\Delta_2 = -1$ и $Q_2 = 3$ (min), т.е. план не оптимален. Разрешающим столбцом является второй, и x_2 будем вводить в базис. Разрешающая строка вторая, и, следовательно, из базиса выведем переменную x_4 . Составляем третью расчетную таблицу (табл. 3.4).

4. При заполнении этой расчетной таблицы добиваемся, чтобы на пересечении разрешающей строки и разрешающего столбца была единица, а остальные элементы a_{2j} разрешающего столбца были равны нулю. Для этого:

- а) третью строку умножаем на 3/2;
- б) из первой и третьей строки вычитаем вторую, добиваясь того, чтобы a_{12} и a_{32} были равны нулю.

В третьей итерационной таблице отрицательных оценок Δ_j нет, следовательно, мы нашли оптимальное решение $x_1 = 7$; $x_2 = 3$ и $\max W = 27$.

3.6. Оптимизация грузопотоков

Задачи оптимизации грузопотоков решаются методами линейного программирования, которые позволяют установить оптимальное закрепление потребителей груза за поставщиками, выбрать маршруты перевозок грузов, решить вопросы распределения парка подвижного состава по АТП и т.д.

В качестве критериев оптимальности принимают пробег подвижного состава, время доставки груза, денежные или трудовые затраты.

3.6.1. Формулировка «транспортной задачи»

Частным случаем задачи линейного программирования является «транспортная задача», которая формулируется следующим образом:

Имеется m поставщиков (A_1, A_2, \dots, A_m), которые располагают запасами однородного груза (Γ) – (a_1, a_2, \dots, a_m) , а также имеется n потребителей (B_1, B_2, \dots, B_n), которым необходимо завести (b_1, b_2, \dots, b_n) тонн груза, а также известны расстояния между грузопунктами.

Необходимо так закрепить поставщиков за потребителями, чтобы затраты на перевозку были \min (т.е. \min транспортной работы).

Исходные данные «транспортной» задачи представляют обычно в виде таблицы 3.5.

Математическая модель транспортной задачи в общем виде записывается в следующей форме.

Дана целевая функция

$$W = \sum_{i=1}^m \sum_{j=1}^n x_{ij} l_{ij} \rightarrow \min \quad (3.11)$$

и ограничения

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = a_i, \quad i = 1, 2, \dots, m \\ \sum_{i=1}^m x_{ij} = b_j, \quad j = 1, 2, \dots, n \end{array} \right\} \quad (3.12)$$

Необходимо найти такие неотрицательные значения переменных $x_{ij} \geq 0$, которые удовлетворяют ограничениям, а целевая функция при этом достигает минимума.

При решении задачи оптимального закрепления потребителей груза за поставщиками символике выражений (3.11) и (3.12) можно придать следующий физический смысл:

m - количество поставщиков;

n - количество потребителей;

a_i - количество груза, отправляемого i -м поставщиком;

b_j - количество груза, доставляемого j -му потребителю;

x_{ij} - количество груза, доставляемого от i -го поставщика j -му потребителю.

L_{ij} - расстояние между i -м и j -м пунктами.

Таблица 3.5

Пункты отправления	Пункты назначения						Запасы a_{ij}
	B_1	B_2	...	B_n			
A_1	l_{11}	l_{12}	...	l_{1n}			
	x_{11}	x_{12}	...	x_{1n}			
A_2	l_{21}	l_{22}	...	l_{2n}			
	x_{21}	x_{22}	...	x_{2n}			
...			
A_m	l_{m1}	l_{m2}	...	l_{mn}			
	x_{m1}	x_{m2}	...	x_{mn}			
Потребности b_j	b_1	b_2	...	b_n			
						$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$	

3.6.2. Последовательность решения транспортной задачи

В решении транспортной задачи выделяют три этапа:

1. Определение первоначального плана перевозок.
2. Проверка полученного плана на оптимальность.
3. Переход к новому плану перевозок с проверкой на оптимальность.

Первый этап. Первоначальный план перевозок принято устанавливать методом северо-западного угла или методом наименьшей стоимости.

Метод северо-западного угла заключается в том, что вначале полностью удовлетворяются потребности первого потребителя (если это возможно) за счет первого поставщика, остаток передается второму потребителю и т.д. Одновременно нужно следить затем, чтобы соблюдался баланс по строкам и столбцам. Этот метод позволяет легко найти первоначальный план перевозок, но он, как правило, далек от оптимального.

Метод наименьшей стоимости предусматривает первоочередное заполнение тех клеток, которые располагают наименьшим расстоянием перевозок (наименьшей стоимостью). При этом план перевозок, как правило, будет ближе к оптимальному.

Второй этап. Проверка первоначального плана на оптимальность может осуществляться различными методами. Наиболее распространенным является метод потенциалов. Согласно данному методу для получения оптимального плана перевозок x_{ij} необходимо и достаточно введение вспомогательных чисел ($U_i, i = 1, 2, \dots, m$ и $V_j, j = 1, 2, \dots, n$, называемых потенциалами, при которых выполнялись бы условия:

$$V_j + U_i = l_{ij}, \quad x_{ij} > 0; \quad (3.13)$$

$$V_j + U_i \leq l_{ij}, \quad x_{ij} = 0, \quad (3.14)$$

где V_j - потенциал j -го столбца, U_i - потенциал i -й строки.

Условие (3.13) должно соблюдаться для занятых клеток таблицы, а условие (3.14) - для свободных. Если условие оптимальности (12) не соблюдается хотя бы для одной клетка, это значит, что полученный план перевозок не является оптимальным и нужно искать другой.

Третий этап решения транспортной задачи (получение нового плана перевозок) рассмотрим на конкретном примере.

3.6.3. Пример решения задачи

Пример. Найти оптимальный план перевозок грузов. Запасы на складах, потребности пунктов назначения и расстояния перевозок однородного груза по пунктам назначения указаны в табл. 3.6.

Решение.

Составляем первоначальный план перевозок методом северо-западного угла (табл. 3.7). Определяем потенциалы по столбцам строкам для занятых клеток по условию (3.13). Один из потенциалов V_j выбираем произвольно.

Например $V_1 = 0$, тогда для клетки $A_1 B_1$ согласно условию

$V_j + U_i = l_{ij}$, имеем $V_1 + U_1 = 4$, если ранее принято $V_1 = 0$, то $U_1 = 4$.

Для клеток $A_2 B_1$ $V_1 + U_2 = 5$ при $V_1 = 0$, $U_2 = 5$;

$A_2 B_2$ $V_2 + U_2 = 10$ при $V_2 = 5$, $U_2 = 5$;

$A_3 B_2$ $V_2 + U_3 = 5$ при $V_2 = 5$, $U_3 = 0$;

$A_3 B_3$ $V_3 + U_3 = 8$ при $V_3 = 8$, $U_3 = 0$;

$A_4 B_3$ $V_4 + U_3 = 6$ при $V_4 = 6$, $U_3 = 0$.

Полученные значения потенциалов проставляем во вспомогательные строки и столбцы расчетной табл. 3.7.

Пункты отправления	Пункты назначения и расстояния перевозок				Запасы a_{ij} , т
	B_1	B_2	B_3	B_4	
A_1	4	8	3	12	100
A_2	5	10	4	18	80
A_3	7	5	8	6	240
Потребности b_j , т	160	90	120	50	420

Таблица 3.6

Таблица 3.7

Пункты отправления	Вспомога- тельные V_j	Пункты назначения и расстояния перевозок				Наличие груза $a_j, т$
		B_1	B_2	B_3	B_4	
	U_i	0	5	8	6	
A_1	4	4 100	9 8	12 3	10 12	100
A_2	5	3 60	10 20	13 -	4 +	18 80
A_3	0	0	5 70	8 120	6 50	240
Потребности $b_j, т$		160	90	120	50	420

Свободные клетки, для которых $x_{ij} = 0$, проверяем на выполнение условия (12), для чего значения суммы потенциалов ($V_j + U_i$) для данных клеток проставляем в верхние левые углы свободных клеток табл. 3.7 и проверяем все эти клетки на выполнение условия $V_j + U_i \leq l_{ij}$.

Сравнением устанавливаем, что для клеток $A_1 B_2$, $A_1 B_3$ и $A_2 B_2$ условие (12) не выполняется, так как соответственно

$$5 + 4 = 9 > 8, \quad 8 + 4 = 12 > 3, \quad 8 + 5 = 13 > 4.$$

Следовательно, план перевозок не оптимален, и его можно улучшить.

Совершенствование плана перевозок осуществляется следующим образом. В клетку, для которой не выполняется условие (3.14), вписываем поставку величиной Δ . Поскольку сумма поставок по строкам и столбцам должна остаться неизменной, то необходимо прибавить и вычесть Δ из поставок в других клетках, обходя их в той последовательности, при которой значение Δ компенсируется вычитанием и сложением со значением числа в клетке. Получаем замкнутую ломаную линию, которую принято называть циклом пересчета.

При получении нового плана перевозок и определении величины поставок Δ необходимо пользоваться следующим правилом: начиная со свободной клетки и двигаясь по циклу пересчета, в вершинах цикла расставляем поочередно знаки $+$ и $-$, затем просматриваем поставки; записанные в отрицательных вершинах, и выбираем наименьшую. Это число прибавляется ко всем поставкам, записанным в положительных вершинах, и вычитается из всех поставок, записанных в отрицательных вершинах. Результаты нового плана перевозок заносим в табл. 3.8.

Таблица 3.8

Пункты отправления	Вспомога- тельные V_j	Пункты назначения и расстояния перевозок				Наличие груза a_j , т			
		B_1 4	B_2 2	B_3 3	B_4 3				
A_1	U_i 0	$\boxed{4}$	4	$\boxed{2}$	8	3	$\boxed{3}$	12	100
A_2	1		3	$\boxed{3}$	10	4	$\boxed{4}$	18	80
A_3	3		7	5	$\boxed{6}$	8		6	240
Потребности b_j , т		160	90	120	50	420			

Вычисление потенциалов табл. 3.8 проведено по условию (11) для занятых клеток. Предварительно принято $U_1 = 0$.

$$V_3 + U_1 = 3 \quad \text{при } V_3 = 3, U_1 = 0;$$

$$V_3 + U_2 = 4 \quad \text{при } V_3 = 3, U_2 = 1;$$

$$V_1 + U_2 = 5 \quad \text{при } V_1 = 4, U_2 = 1;$$

$$V_1 + U_3 = 7 \quad \text{при } V_1 = 4, U_3 = 3;$$

$$V_4 + U_3 = 6 \quad \text{при } V_4 = 3, U_3 = 3.$$

Для всех свободных клеток табл. 8.8 условие $V_j + U_i \leq l_{ij}$ выполняется, следовательно, получен оптимальный план перевозок. Значение целевой функций для такого плана перевозок таково:

$$W = 3 \cdot 100 + 5 \cdot 60 + 4 \cdot 20 + 7 \cdot 100 + 5 \cdot 90 + 6 \cdot 50 = 2130 \text{ (тыс. км.)}$$

Примечание. Выше рассмотрен вариант транспортной задачи в которой потребности в грузах пунктов назначения равны запасу груза в пунктах отправления;

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (3.15)$$

Модель такой транспортной задачи называется закрытой. Если же условие (8.15) выполняется, то модель такой задачи называется открытой.

В случае превышения запаса над потребностью, т.е., $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ при решении задачи вводится фиктивный $(n + 1)$ -й пункт назначения с

потребностью $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$ и соответствующее расстояние (тариф), равное нулю $l_{i,n+1} = 0$, ($i = 1, 2, \dots, m$).

Аналогично при $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$, вводится фиктивный ($m + 1$)-й пункт отправления с запасом груза $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$, и расстояния (тарифы) полагаются равными нулю: $l_{m+1,j} = 0$, ($j = 1, 2, \dots, n$). Этим задача сводится к обычной (закрытой) транспортной задаче.

3.7. Разработка рациональных маршрутов перевозок массовых грузов на основании заявок договорной клиентуры

Составление рациональных маршрутов перевозок массовых грузов имеет целью достижение максимального коэффициента использования пробега автомобилей путем определения порядка следования автомобилей от места разгрузки к месту погрузки с тем условием, чтобы общий пробег без груза был наименьшим.

3.7.1. Постановка задачи

При решении таких задач отбираются те заявки на перевозку грузов, которые совпадают по времени выполнения перевозок и которые можно осуществить одним и тем же подвижным составом.

Предположим, что такие заявки отображены, установлены кратчайшие расстояния между всеми пунктами, выбран тип подвижного состава, определено количество поездок по доставке груза от каждого поставщика соответствующему потребителю. Исходные данные одного из вариантов подобной задачи представлены табл. 8.9, 8.10, где соответственно отражены суточный объем перевозок по заявкам грузоотправителей и показатели работы автомобилей.

Таблица 3.9

№ п/п	Грузоотправитель		Грузополучатель		Род груза	Количество	
						т	ездки
1	Речной порт	A ₁	АБЗ	Б ₃	Песок	140	20
2	Речной порт	A ₁	ЗЖБК	Б ₄	Щебень	105	15
3	Котлован	A ₂	Строительство 11	Б ₂	Грунт	245	35
4	Песчаный карьер	A ₃	Строительство 1	Б ₁	Песок	336	48

Каждому отправителю присвоено условное обозначение А, потребителю - Б с соответствующими порядковыми цифровыми индексами.

В табл. 3.9 кроме количества перевозимого груза показано количество ездов, которое определено по показателям работы выбранного подвижного состава (табл. 3.10).

Таблица 3.10

№ п/п	Параметр	Единица измерения	Количество
1	Грузоподъемность МАЗ – 503Б	т	7,0
2	Средняя техническая скорость	км/ч	22,0
3	Время в наряде	ч	14,0
4	Норма времени на погрузку за езду	мин	7,0
5	Норма времени на разгрузку за езду	мин	6,0
6	Начало работы пунктов погрузки	ч	7,0

Матрица расстояний между грузопунктами, соответствующая схеме перевозок (рис. 3.6), представлена в табл. 3.11.

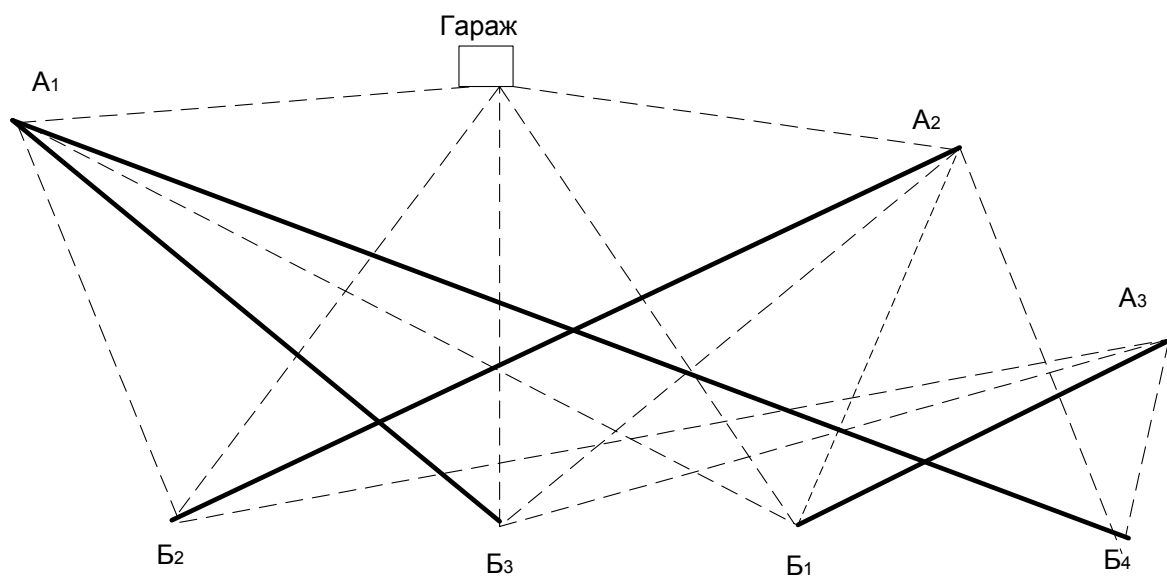


Рис. 3.6. Схема перевозок

Таблица 3.11

Грузопункты	А ₁	А ₂	А ₃	АТП
Б ₁	12	13	11	10
Б ₂	9	12	13	9
Б ₃	11	15	12	11
Б ₄	14	8	5	9
АТП	6	4	7	0

3.7.2. Разработка рациональных маршрутов

На основании заявок договорной клиентуры заполняют расчетную таблицу (например методом северо-западного угла), в клетках которой указывают количество порожних ездов из пунктов B_j в пункты A_i , и расстояние между пунктами (последние проставляются в верхних правых углах соответствующих клеток). Получим первоначальный план холостых ездов. Проверим этот план на оптимальность, например методом потенциалов, и добьемся такого распределения порожних ездов автомобилей, при которых их пробег без груза будет минимальным (см. метод решения транспортной задачи).

Результаты расчета оптимального распределения ездов автомобилей без груза, обеспечивающих минимальный порожний пробег всех автомобилей, участвующих в планируемых перевозках, приведены в табл. 3.12. В данной таблице указано, что из пункта B_2 в пункт A необходимо сделать 35 ездов без груза, из пункта B_3 в пункт A_3 - 20 ездов и т.д.

Таблица 3.12

Потребитель	Вспомогательные	Пункты назначения и расстояния перевозок			Количество ездов	
		A_1	A_2	A_3		
	U_j	0	3	0		
	V_i					
B_3	12	12 (20)	11 (10)	15 20	12 20	20
B_4	5	5 (15)	14 (7)	8 0	5 15	15
B_2	9	35	9 (35)	12 0	13 (9)	35
B_1	10	10	17	13 35	13 10 (48)	48
					11	
Количество ездов		35	35	48		118

После получения оптимального плана распределения порожних ездов в эту же таблицу вносим план груженых ездов (цифры в скобках).

В тех клетках, где имеются две цифры, назначаются маятниковые маршруты, количество ездов по которым равно наименьшей цифре. Так, в клетке $A_3 B_1$ имеем маятниковый маршрут №1: $A_3 B_1 B_1 A_3$ с 13 оборотами. Это количество ездов исключается из дальнейшего рассмотрения.

Когда все маятниковые маршруты найдены, в матрице строим четырехугольные (шестиугольные и т.д.) контуры, все вершины которых лежат в загруженных клетках, причем вершины с грузеными езтками должны чередоваться с вершинами порожних ездов. В данном примере получено два таких контура, которые показаны пунктирными линиями. Каждый из них составляет кольцевой маршрут.

Маршрут №2:

$A_3 B_1 B_1 A_2 A_2 B_2 B_2 A_1 A_1 B_3 B_3 A_3$ - 20 оборотов;

Маршрут №3:

$A_3 B_1 B_1 A_2 A_2 B_2 B_2 A_1 A_1 B_4 B_4 A_3$ - 15 оборотов.

Количество оборотов на маршруте определяется наименьшим числом в вершинах контура. Выбранное количество ездов из клеток таблицы исключается. Решение ведется до полного исключения всего количества ездов из матрицы.

После назначения маршрутов необходимо выбрать первоначальные пункты погрузки на кольцевых маршрутах и определить, какое количество автомобилей следует направить на каждый маршрут, чтобы обеспечить выполнение плана перевозок.

Первоначальный пункт погрузки выбирается из условия минимума нулевого пробега l_{0A_i} . Для нашего примера минимальный нулевой пробег $l_{0A_2} = l_{ГА_2} + l_{B_{11}} = 4 + 10 = 14$ км для первоначального пункта погрузки A_2 . Схема кольцевого маршрута №3 приведена на рис. 3.7.

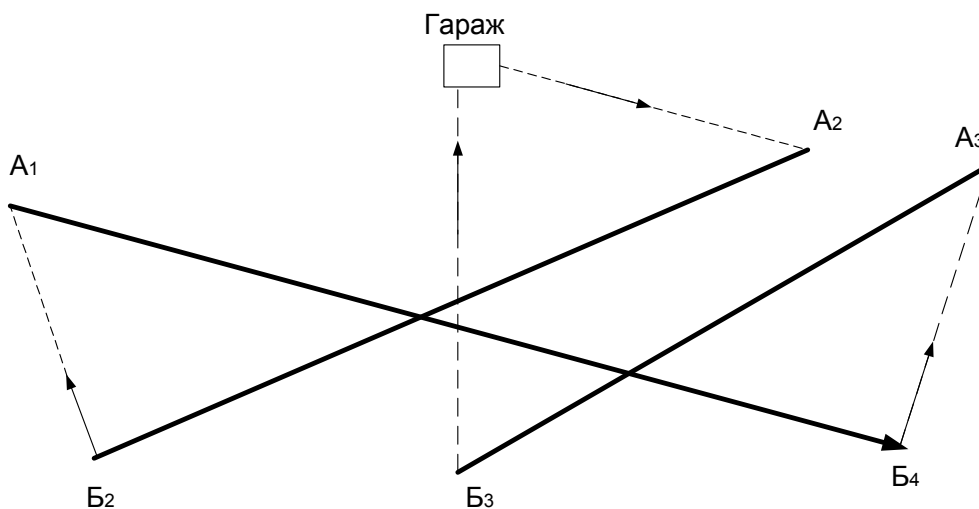


Рис. 3.7. Схема кольцевого маршрута

Перевозку по маршруту можно начинать из пункта A_1 или A_3 , тогда автомобиль будет возвращаться в гараж из пункта B_2 или B_4 соответственно. Это приходится часто делать с целью исключения одновременной подачи на один и тот же пункт погрузки большого количества автомобилей и создания очередей, хотя при этом нулевые пробеги автомобилей могут несколько увеличиваться.

3.7.3. Расчет потребного числа подвижного состава на маршруте

Расчет потребного числа автомобилей на маршруте и коэффициента использования парка проводится по ниже приведенным соотношениям.

Число оборотов автомобилей по маршруту за время в наряде

$$Z_{\text{об.}} = \frac{T_{\text{н}} - (l_{01} + l_{02} - l'_x) / V_{\text{т}}}{t_{\text{об}}},$$

где T - время в наряде, ч; l_{01} - первый нулевой пробег, км; l_{02} - второй нулевой пробег, км; l'_x - последняя холостая ездка на маршруте, км; $V_{\text{т}}$ - средняя техническая скорость, км/ч; $t_{\text{об}}$ - время оборота автомобиля на маршруте, ч.

$$t_{\text{об}} = \frac{2l_{\text{е.г.}}}{V_{\text{т}}} + t_{\text{пр.}} - \text{маятниковый маршрут};$$

$$t = \frac{l_{\text{м}}}{V_{\text{т}}} + \sum_{i=1}^n t_{\text{пр.}i} - \text{кольцевой маршрут},$$

где $l_{\text{е.г.}}$ - расстояние ездки с грузом, км; $t_{\text{пр.}}$ - время погрузочно-разгрузочных работ за ездку, ч; $l_{\text{м}}$ - длина маршрута, км; n - число ездок за оборот.

Потребное число автомобилей на маршруте

$$A_{\text{сут.}} = \frac{U_{\text{сут.}}}{Q_{\text{сут.}}} = \frac{U_{\text{сут.}}}{q_{\text{н}} \gamma Z_{\text{об.}} n},$$

где $U_{\text{сут.}}$ - плановый объем перевозок на маршруте за сутки, т; $q_{\text{н}}$ - номинальная грузоподъемность, т; γ - коэффициент использования грузоподъемности.

Коэффициент использования пробега за день

$$\beta_{\text{р.д.}} = \frac{L_{\text{гр.}}}{L_{\text{об.}}},$$

где $L_{\text{гр.}}$ - пробег с грузом автомобиля за день, км.

$$L_{\text{гр.}} = Z_{\text{об.}} \Sigma l_{\text{ег}i}; L_{\text{об.}} = Z_{\text{об.}} l_{\text{м}} + \sum_{i=1}^n l_{0i} - l_x,$$

где $\Sigma l_{\text{ег}i}$ - суммарный пробег с грузом за один оборот автомобиля на маршруте; Σl_{0i} - суммарный нулевой пробег, км; $l_{\text{м}}$ - длина маршрута.

Контрольные вопросы

1. Сформулируйте задачу линейного программирования.
2. Запишите задачу линейного программирования в матричной форме и в виде знаков суммирования.
3. Что такое область допустимых решений, опорные точки и опорные решения?
4. Перечислите правила перехода от исходной задачи линейного программирования к двойственной задаче.
5. Последовательность решения задачи линейного программирования геометрическим способом.
6. Особенности решения задач линейного программирования симплексным методом.
7. Что такое каноническая форма записи задачи линейного программирования?
8. Последовательность решения задачи линейного программирования симплекс-методом.
9. Сформулируйте «транспортную задачу».
10. Последовательность решения «транспортной задачи».
11. Особенности записи и решения открытой «транспортной задачи».

Список литературы

1. Вентцель Е.С. Исследование операций. М.: Высшая школа, 1980.
2. Сборник задач по математике для вузов «Методы оптимизации». Под ред. Ефимова А.В. 1990.
3. Коновалов С.И., Савин В.В. Моделирование случайных процессов автомобильного транспорта, I и II части. Владимир, ВлГУ, 2004, 2005 г.г.
4. Коновалов С.И. Моделирование производственных процессов автомобильного транспорта. Учебн. пособие / Владим. гос. ун-т; Сост. С.И. Коновалов, С.А. Максимов, В.В. Савин. Владимир, 2005, - 244 с.
5. Оптимизационное моделирование производственных процессов на предприятиях автомобильного транспорта : метод. указания к лабораторным работам / Владим. гос. ун-т имени Александра Григорьевича и Николая Григорьевича Столетовых ; сост. И. В. Денисов. – Владимир : Изд-во ВлГУ, 2012. – 72 с.