

ГЛАВА 9. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Динамическое программирование – это метод, который может быть использован для решения различных оптимизационных проблем, которые предполагают возможность разворачивания их в пространстве и времени. Во многих приложениях динамическое программирование приносит результат, развивая процесс решения от конца к началу, а также разбивая большую громоздкую проблему на серию более мелких, легко поддающихся решению задач.

Среди классических задач динамического программирования рассматриваются задачи распределения капитальных вложений, замены оборудования, набора высоты самолетом, нахождения кратчайшего расстояния на сети, определения оптимальной структуры мероприятий по повышению эффективности производства и некоторые другие. В данной теме будут рассмотрены две из них как иллюстрация метода, и учитывающие два способа построения процедуры продвижения по этапам решения: от конца к началу и от начала к концу.

9.1. ДВЕ ГОЛОВОЛОМКИ

Рассмотрим, как можно решить, казалось бы, сложные задачи, продвигаясь от конца решения к началу и приводя их к тривиальному решению.

Пример 1.

Допустим, что на столе лежат 30 спичек. Я начинаю игру, забирая со стола 1, 2 или 3 спички. Затем мой противник также должен взять 1, 2 или 3 спички. Мы продолжаем до тех пор, пока последняя спичка не будет взята со стола. Игрок, который берет последнюю спичку, считается проигравшим. Как я, начинающий игру первым, могу быть уверенным в том, что выиграю игру?

Решение.

Если я буду уверен в том, что, когда наступит черед моего противника брать спички, на столе останется одна спичка, то я выиграю игру. Вернувшись на один шаг назад, я могу быть уверен, что перед тем, как мой противник будет брать со стола спички, на столе будут лежать 5 спичек, то я также выиграю игру. Дело в том, что независимо от того, сколько спичек он возьмет, я смогу взять столько, что на столе перед его ходом останется 1 спичка.

Аналогично, если я смогу «заставить» оппонента играть, когда на столе будет 5, 9, 13, 17, 21, 25 или 29 спичек, то победа мне будет обеспечена. Так, я не проиграю, если первым ходом я возьму 1 спичку. Затем я должен лишь следить, чтобы противник остановился перед выбором, когда на столе лежат 29, 25, 21, 17, 13, 9 или 5 спичек.

Таким образом, рассмотренная проблема легко решается от конца к началу. Это гораздо сложнее сделать, двигаясь от начала к концу.

Пример 2.

У меня две емкости для жидкости: одна 9-литровая, а другая 4-литровая. Мне необходимо иметь ровно 6 литров жидкости. Как достичь этой цели, используя имеющиеся в наличии емкости?

Решение.

Начиная почти в самом конце проблемы, я четко представляю, что если в 4-литровой емкости окажется 1 литр жидкости, то я легко смогу решить задачу. Тогда я смогу налить в 9-литровую емкость 9 литров жидкости и 3 из них перелить в 4-литровую (в которой уже имеется 1 литр). Тогда в 9-литровой останется 6 литров. После такого «прозрения», ход решения может быть представлен в таблице (начальная ситуация записана последней, а конечная первой):

Количество литров в 9-литровой емкости	Количество литров в 4-литровой емкости
6	0
6	4
9	1
0	1
1	0
1	4
5	0
5	4
9	0
0	0

9.2. ЗАДАЧИ НА СЕТИ

Большое количество приложений динамического программирования сокращает множество точек для нахождения кратчайшего (или наиболее длинного) пути, объединяя две соседние точки на заданной сети. Рассмотрим, как движение в обратном направлении позволяет решить проблему кратчайшего пути.

Пример 3.

Петр живет во Львове. Однако ему там не везет, и он решил предпринять путешествие в Донецк в поисках судьбы и удачи. Он планирует проехать все расстояние на автомашине. Причем, учитывая свое финансовое положение, Петр решает так разработать свой маршрут, чтобы в конце каждого дня пути останавливаться на ночлег у друзей, которые живут в Ровно, Виннице, Черновцах, Киеве, Кировограде, Николаеве, Запорожье и Полтаве.

Петр знает, что после 1-го дня пути он может добраться либо до Ровно, либо до Винницы, либо до Черновцов. После 2-х дней пути он может достичь Киева, Кировограда или Николаева. К концу 3-го дня он может быть в Запорожье или Полтаве. И наконец, после 4-х дней пути, он может достичь Донецка. Где должен Петр ночевать в конце каждого дня пути, чтобы минимизировать общее расстояние? В таблице приведены действительные расстояния между городами.

Решение.

Начнем решение задачи, двигаясь от конца к началу. На рис. 9.1 все города классифицированы в зависимости от номера дня, когда Петр сможет прибыть туда. Дни приняты в качестве стадий процесса решения задачи. Считая, что в начале 1-го дня Петр покинет Львов, а Ровно, Винницу или Черновцы он покинет в начале 2-го дня и т.д.

Города	Расстояние
1-2	215
1-3	369
1-4	278
2-5	324
2-6	618
2-7	779
3-5	266
3-6	317
3-7	466
4-5	538
4-6	637
4-7	692
5-8	343
5-9	568
6-8	251
6-9	314
7-8	488
7-9	352
8-10	391
9-10	243

Начнем с поиска кратчайшего пути в Донецк из каждого из городов, которые расположены в одном дне пути от Донецка. Имея эту информацию, мы сможем найти кратчайшее расстояние от городов, что в 2-х днях пути от Донецка и т.д.

Наконец, мы найдем кратчайшее расстояние от города, который в 4-х днях пути от Донецка, т.е. от Львова. Для простоты объяснения, каждому городу поставлено в соответствие число, начиная с 1 (Львов) и заканчивая 10 (Донецк).

Пусть c_{ij} – расстояние в км от города i до города j . Так, например, $c_{35} = 266$. Это расстояние от Винницы до Киева.

Определим также через $f_i(i)$ – кратчайшее расстояние от города i до Донецка, принимая во внимание, что город i принадлежит t -й стадии.

4-я стадия расчетов.

Сначала найдем кратчайшее расстояние от каждого города 4-й стадии до Донецка.

Так как имеется всего лишь по одному пути от каждого города 4-й стадии до Донецка, мы можем сказать, что $f_4(8) = 391$ – это кратчайший путь от Полтавы, а $f_4(9) = 243$ – кратчайший (и единственный) путь от Запорожья до Донецка.

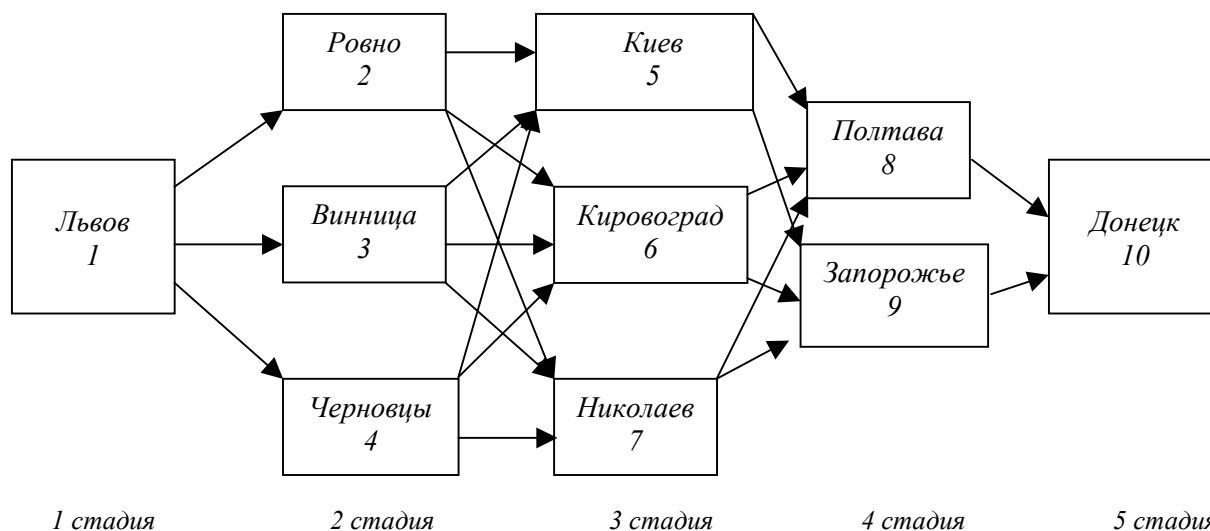


Рис. 9.1. Схема маршрутов движения между городами

3-я стадия расчетов.

Переместимся еще на одну стадию назад и определим кратчайшее расстояние от городов 3-й стадии до Донецка. Так, например, для 5-го города возможны 2 пути:

1-й путь. Поехать из 5-го в 8-й город и затем выбрать кратчайший путь из 8-го города в 10-й.

2-й путь. Поехать из 5-го в 9-й город и затем выбрать кратчайший путь из 9-го города в 10-й.

Длина 1-го пути определяется следующим образом: $c_{58} + f_4(8)$, длина 2-го пути: $c_{59} + f_4(9)$.

Отсюда, кратчайший путь от 5-го города до 10-го может быть записан следующим образом:

$$f_3(5) = \min \begin{cases} c_{58} + f_4(8) = 343 + 391 = 734 \\ c_{59} + f_4(9) = 568 + 243 = 811. \end{cases}$$

Таким образом, из города 5 в Донецк кратчайшим путем является путь 5–8–10. Чтобы получить этот результат, мы должны знать $f_4(8)$ и $f_4(9)$.

Аналогично находим $f_3(6)$ – кратчайший путь от 6-го города до Донецка, учитывая, что мы должны поехать либо в город 8, либо в город 9:

$$f_3(6) = \min \begin{cases} c_{68} + f_4(8) = 251 + 391 = 642 \\ c_{69} + f_4(9) = 314 + 243 = 557. \end{cases}$$

Итак, $f_3(6) = 557$ и наиболее короткий путь из города 6 в город 10 лежит через пункты 6–9–10.

Найдем
$$f_3(7) = \min \begin{cases} c_{78} + f_4(8) = 488 + 391 = 879 \\ c_{79} + f_4(9) = 352 + 243 = 595. \end{cases}$$

Следовательно, $f_3(7) = 595$ и кратчайший путь из 7 в 10 – это путь, проходящий через города 7–9–10.

2-я стадия расчетов.

Зная $f_3(6)$ и $f_3(7)$, очень легко вернуться на один шаг и рассчитать $f_2(2)$, $f_2(3)$ и $f_2(4)$, т.е. кратчайшие пути из 2-го, 3-го и 4-го городов до Донецка. Так, самый короткий путь из города 2 должен пролегать либо через город 5, либо через город 6, либо через город 7. Так как мы должны попасть в эти города, то дальнейший путь должен пролегать по самому короткому маршруту.

1-й путь. Едем из города 2 в город 5. Затем следуем кратчайшим путем в город 10: $c_{25} + f_3(5)$;

2-й путь. Едем из города 2 в город 6. Затем следуем кратчайшим путем в город 10: $c_{26} + f_3(6)$;

3-й путь. Едем из города 2 в город 7. Затем следуем кратчайшим путем в город 10: $c_{27} + f_3(7)$.

Таким образом, мы можем рассчитать кратчайший путь из города 2 в город 10.

$$f_2(2) = \min \begin{cases} c_{25} + f_3(5) = 324 + 734 = 1058 \\ c_{26} + f_3(6) = 618 + 557 = 1175 \\ c_{27} + f_3(7) = 779 + 595 = 1374. \end{cases}$$

Следовательно, $f_2(2) = 1058$ и кратчайший путь из 2-го города в 10-й город пролегает через города 2–5–8–10.

$$\text{Аналогично } f_2(3) = \min \begin{cases} c_{35} + f_3(5) = 266 + 734 = 1000 \\ c_{36} + f_3(6) = 317 + 557 = 874 \\ c_{37} + f_3(7) = 466 + 595 = 1061. \end{cases}$$

Итак, $f_2(3) = 874$ и кратчайший путь проходит через города 3–6–9–10.

$$\text{Наконец, } f_2(4) = \min \begin{cases} c_{45} + f_3(5) = 538 + 734 = 1272 \\ c_{46} + f_3(6) = 637 + 557 = 1194 \\ c_{47} + f_3(7) = 642 + 595 = 1237. \end{cases}$$

Отсюда, $f_2(4) = 1194$. Кратчайший путь из 4-го в 10-й город: 4–6–9–10.

1-я стадия расчетов.

Имея информацию относительно $f_2(2)$, $f_2(3)$ и $f_2(4)$, мы можем, вернувшись еще на один шаг, определить кратчайший путь из 1-го города в 10-й. Он может пройти через 2-й, 3-й или через 4-й город. Это означает, что кратчайший путь из Львова в Донецк может быть одним из следующих:

1-й путь. Едем из 1-го города во 2-й и затем выбираем кратчайший путь в 10-й город: $c_{12} + f_2(2)$;

2-й путь. Едем из 1-го города в 3-й и затем выбираем кратчайший путь в 10-й город: $c_{13} + f_2(3)$;

3-й путь. Едем из 1-го города в 4-й и затем выбираем кратчайший путь в 10-й город: $c_{14} + f_2(4)$.

Исходя из этого, находим $f_1(1)$

$$f_1(1) = \min \begin{cases} c_{12} + f_2(2) = 215 + 1058 = 1273 \\ c_{13} + f_2(3) = 369 + 874 = 1243 \\ c_{14} + f_2(4) = 278 + 1194 = 1472. \end{cases}$$

Итак, $f_1(1) = 1243$. Кратчайший путь пролегает через города 1–3–6–9–10, что означает, что из Львова Петр должен ехать в Винницу, затем в Кировоград, потом в Запорожье и, наконец, в Донецк. Общая длина пути составит 1243 км.

9.3. ВЫЧИСЛИТЕЛЬНАЯ ЭФФЕКТИВНОСТЬ МЕТОДА ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Рассмотренный пример не является характерным с точки зрения иллюстрации эффективности алгоритма динамического программирования, так как в нем (в примере) необходимо было сравнить длины всего 18 путей ($3 \times 3 \times 2$). Однако для

более громоздких сетей динамическое программирование является более эффективным методом нахождения кратчайшего пути.

Пусть имеется граф, в котором 27 вершин (см. рис. 9.2).

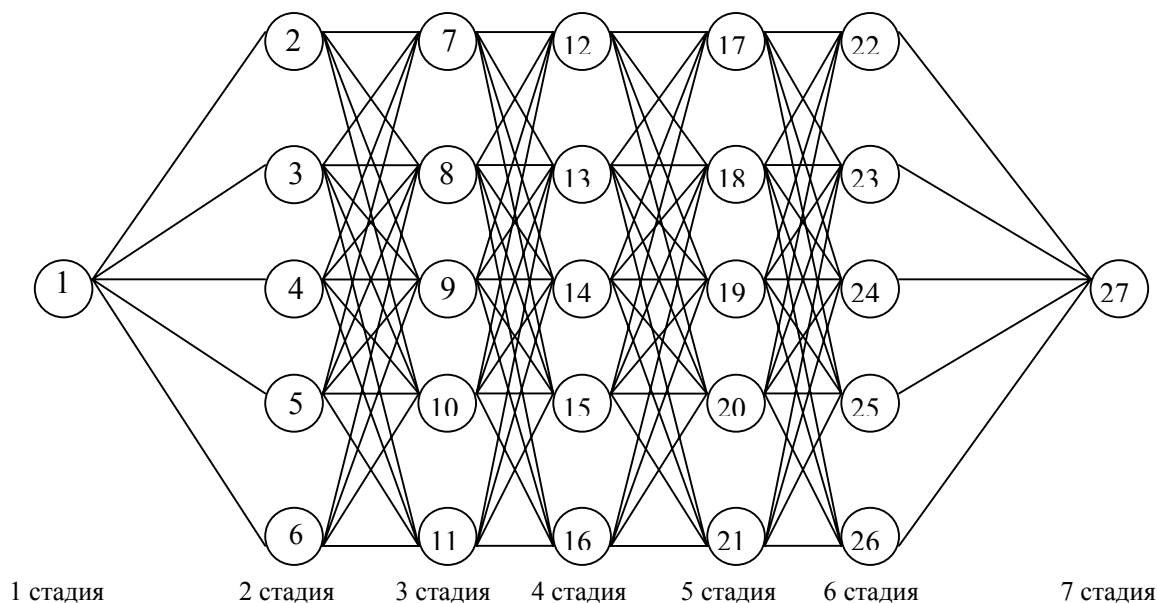


Рис. 9.2. Иллюстративный пример

Простое перечисление длин возможных путей потребует $5^5 \times 5 = 15625$ операций сложения.

Используя метод динамического программирования и, начиная процесс решения с конца, мы определяем $f_6(22), f_6(23), f_6(24), f_6(25)$ и $f_6(26)$ – кратчайшие пути от пунктов 6-й стадии до 27-го пункта. Эти вычисления не требуют действий сложения.

Затем находим $f_5(17), f_5(18), f_5(19), f_5(20)$ и $f_5(21)$. Так, например, для нахождения $f_5(21)$ мы используем выражение

$$f_5(21) = \min_j \{c_{21,j} + f_6(j)\}, \quad j = 22 - 26.$$

Вычисления требуют 5-ти действий сложения. Для нахождения всех значений кратчайших расстояний на 5-й стадии требуются $5 \times 5 = 25$ действий сложения. Аналогично, для 4-й стадии необходимо 25 действий сложения. Для 3-й и 2-й стадий аналогично. Для первой стадии необходимо всего 5 раз выполнить эти действия. Итак, используя метод динамического программирования применительно к данной задаче, необходимо произвести $4 \times 25 + 5 = 105$ действий сложения. В сравнении с простым перечислением это незначительное число (в 148.8 раз меньше). По количеству длин пути преимущество также на стороне метода динамического программирования: 3124 против 84

$$3124 = 5^5 - 1,$$

$$84 = 20 \cdot (5 - 1) + 4.$$

9.4. ОСОБЕННОСТИ МЕТОДА ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Для большинства приложений метода динамического программирования имеется ряд общих особенностей

1-я особенность. *Проблема требует разбиения на стадии с необходимостью принятия решения на каждой стадии (хотя это бывает и не всегда).*

Так, в нашем примере каждая стадия t состоит из тех городов, в которых может оказаться Петр в начале t -го дня своего путешествия.

2-я особенность. *Каждая стадия имеет определенное число состояний, ассоциирующихся с ней. Под состоянием будем понимать информацию, которая необходима на каждой стадии для принятия оптимального решения.*

В нашем примере состояния – это конкретные города, в которые может попасть Петр в начале t -го дня. Так, для стадии 3 возможными состояниями могут быть Киев, Кировоград и Николаев.

Следует заметить, что для того, чтобы сделать правильное решение на любой стадии, Петру нет необходимости знать, как он попал в свое настоящее местонахождение.

Так, если Петр в Киеве, то остальные решения не зависят от того, как он попал в Киев. Его будущее решение зависит лишь от факта, что он сейчас находится в Киеве.

3-я особенность. *Решение, выбранное на любой стадии, показывает, как состояние на данной стадии трансформируется в состояние на следующей стадии.*

В нашем примере решение Петра на любой стадии – определить, в какой следующий город он попадет.

В ряде случаев, однако, принимаемое решение не дает информации о следующем состоянии с определенностью. Напротив, текущее решение может лишь определить закон распределения состояний на следующей стадии.

4-я особенность. *Для каждого конкретного состояния оптимальное решение на каждой из оставшихся стадий не должно зависеть от ранее достигнутых состояний или ранее принятых решений. Эта идея известна как **принцип оптимальности**.*

Данная особенность дает общее понимание **принципа оптимальности Ричарда Беллмана**. Более полно он формулируется следующим образом:

Каково бы ни было начальное состояние и принятое начальное решение, каждое последующее решение должно быть оптимальным относительно состояния, полученного в результате предыдущего решения.

В контексте нашей проблемы принцип оптимальности сводится к следующему.

Предположим, что кратчайший путь (пусть это будет R) из города 1 в город 10 должен пройти через город i (это известно точно). Тогда часть пути, что идет из города i в город 10, должна быть кратчайшим путем из i в 10.

Если бы это было так, то мы могли бы найти путь из города 1 в город 10, который был бы короче, чем R , присоединяя кратчайший путь из i в 10 к части R , которая ведет из 1 в i . Это бы позволило создать путь из 1 в 10, который короче, чем R , тем самым, противореча тому факту, что R – самый короткий путь.

Например, если известно, что кратчайший путь из города 1 в город 10 протекает через город 2, то кратчайший путь из 1 в 10 должен включать кратчайший путь из 2 в 10, т.е. 2–5–8–10. Отсюда следует, что любой путь из 1 в 10, который проходит через пункт 2 и не содержит кратчайшего пути из 2 в 10, имеет протяженность больше, чем кратчайший путь: $c_{12} + (\text{нечто большее, чем } f_2(2))$. Следовательно, такой путь не может быть короче кратчайшего пути.

5-я особенность.

Если состояния проблемы были классифицированы так, что они попали под одну из T стадий, то должно быть выражение, которое соотносит затраты или доход, получаемые на стадиях $t, t+1, \dots, T-1$ с затратами или доходом, получаемым на стадиях $t+1, t+2, \dots, T$.

Иными словами, это соотношение формирует процедуру движения обратным ходом (называется **рекуррентное соотношение**).

В нашем примере это можно записать следующим образом:

$$f_t(i) = \min_j (c_{ij} + f_{t+1}(j)),$$

где j должно быть состоянием $t+1$, а

$$f_5(10) = 0.$$

Теперь можно описать процедуру, как принимать оптимальное решение.

Предположим, что начальное состояние на стадии 1 – i_1 . Чтобы использовать рекуррентное соотношение, мы начинаем с нахождения оптимального решения для каждого состояния, ассоциирующегося с последней стадией. Затем используем рекуррентное соотношение, описанное в 5-й особенности для определения $f_{T-1}(\bullet)$ (вместе с оптимальным решением) для каждого состояния стадии $T-1$. Повторяем действие для стадии $T-2$. И так далее, пока не получим $f_1(i_1)$ и оптимальное решение, находясь на стадии 1 в состоянии i_1 .

Тогда наше оптимальное решение на стадии 1 выбирается из серии решений, соответствующих $f_1(i_1)$.

Выбранное на стадии 1 решение приведет нас к некоторому состоянию стадии 2 (пусть это будет i_2). Затем, на стадии 2 выбираем решение, соответствующее $f_2(i_2)$. Действие продолжаем таким образом, пока решение не будет принято для каждой стадии.

Существует также альтернативный метод, позволяющий строить рекуррентные соотношения, двигаясь от начала процесса к его завершению.

9.5. ЗАДАЧА ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОЙ СИСТЕМЫ МЕРОПРИЯТИЙ ПО РОСТУ ПРОИЗВОДИТЕЛЬНОСТИ ТРУДА

В ранее рассмотренной задаче о горе-путешественнике процесс ее решения разворачивался от конца к началу. Рассмотрим подход, согласно которому схема расчетов диаметрально противоположна – от начала к концу.

Пример 4.

На промышленном предприятии намечается внедрить систему мер по повышению эффективности производства и росту производительности труда. В настоящее время на предприятии работает 960 человек. Предполагается, что объем выпуска продукции возрастет на 2%. Данный прирост намечается обеспечить за счет роста производительности труда на 3.5%. Для достижения заданных показателей деятельности предприятия разработана система мероприятий. Причем по каждому направлению их внедрения имеются альтернативные варианты, отличающиеся показателями затрат и условным высвобождением численности работников (как показателя, отражающего рост производительности труда). Указанные мероприятия приведены в таблице.

В таблице использованы следующие обозначения: B – условное высвобождение численности работающих, чел.; K – затраты на реализацию мероприятия, тыс. грн.

Необходимо определить оптимальное сочетание вариантов мероприятий, обеспечивающее достижение заданных показателей с минимальными затратами.

Вариант	Мероприятия									
	1		2		3		4		5	
	B	K	B	K	B	K	B	K	B	K
1	2	8	5	14	5	11	8	11	8	18
2	3	11	7	18	6	13	9	15	10	24
3					7	17			11	30

Решение.

Прежде чем решать задачу, определим, какому условному высвобождению численности работников соответствует рост производительности труда 3.5% с учетом роста объема производства на 2%. Для этого воспользуемся формулой

$$B_{пл.} = \frac{A_{пл.} \cdot \alpha_{баз.} \cdot \beta}{A_{баз.} (100 + \beta)}$$

или

$$B_{пл.} = \left(1 + \frac{\alpha}{100}\right) \times \frac{\alpha_{баз.} \cdot \beta}{(100 + \beta)},$$

где $B_{пл.}$ – необходимая численность условно высвобождаемых, чел.;

$A_{пл.}$ – объем производства на плановый период, млн. грн.;

$A_{баз.}$ – объем производства в отчетном периоде, млн. грн.;

$Ч_{баз.}$ – численность работающих в отчетном периоде, чел.;

α – рост объема производства, %;

β – рост производительности труда, %.

Подставляя исходные данные, получаем

$$B_{пл.} = \left(1 + \frac{2}{100}\right) \times \frac{960 \times 3.5}{(100 + 3.5)} = 33 \text{ чел.}$$

С целью построения математической модели введем переменные. Пусть x_{ij} – булева переменная, определяющая принятие ($x_{ij} = 1$) или непринятие ($x_{ij} = 0$) к реализации j -го варианта i -го мероприятия.

Используя информацию из таблицы, построим математическую модель. Цель решения задачи – минимизация затрат на реализацию мероприятий по росту производительности труда

$$\begin{aligned} \min Z = & 8x_{11} + 11x_{12} + 14x_{21} + 18x_{22} + 11x_{31} + 13x_{32} + 17x_{33} + \\ & + 11x_{41} + 15x_{42} + 18x_{51} + 24x_{52} + 30x_{53}. \end{aligned}$$

При этом должны быть соблюдены следующие ограничения:

1) по заданному условному высвобождению численности работающих

$$\begin{aligned} & 2x_{11} + 3x_{12} + 5x_{21} + 7x_{22} + 5x_{31} + 6x_{32} + 7x_{33} + \\ & + 8x_{41} + 9x_{42} + 8x_{51} + 10x_{52} + 11x_{53} = 33, \end{aligned}$$

2) по принятию только одного варианта каждого мероприятия

$$\begin{aligned} x_{11} + x_{12} &= 1, \\ x_{21} + x_{22} &= 1, \\ x_{31} + x_{32} + x_{33} &= 1, \\ x_{41} + x_{42} &= 1, \\ x_{51} + x_{52} + x_{53} &= 1, \end{aligned}$$

3) на условия изменения переменных

$$\begin{aligned} x_{11} &= 0 \vee 1, x_{12} = 0 \vee 1, x_{21} = 0 \vee 1, x_{22} = 0 \vee 1, \\ x_{31} &= 0 \vee 1, x_{32} = 0 \vee 1, x_{33} = 0 \vee 1, x_{41} = 0 \vee 1, \\ x_{42} &= 0 \vee 1, x_{51} = 0 \vee 1, x_{52} = 0 \vee 1, x_{53} = 0 \vee 1. \end{aligned}$$

Опишем кратко ход решения задачи методом динамического программирования. Весь процесс будет разбит на стадии в соответствии с числом мероприятий.

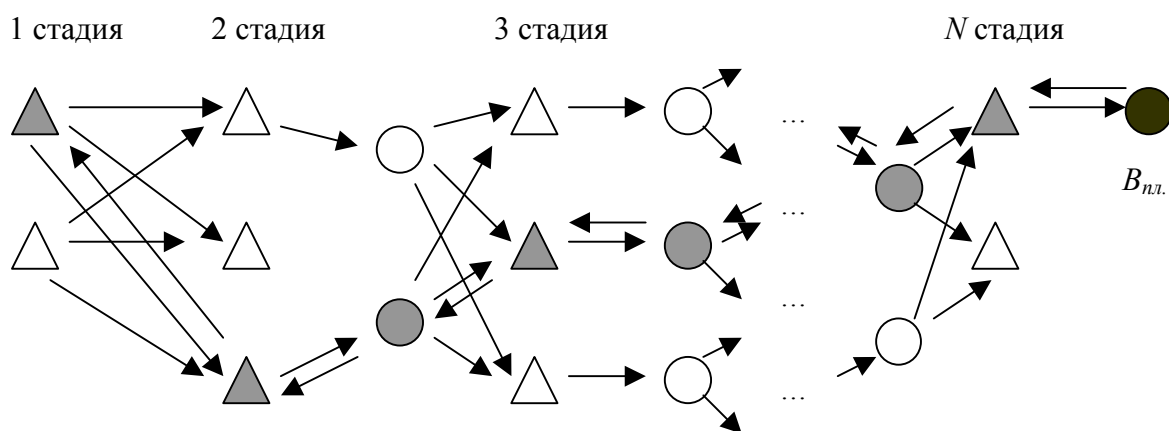
На первой стадии осуществим оценку альтернативных вариантов первого мероприятия по затратам. В случае, если встретится ситуация, когда меньшему числу высвобождаемых соответствуют большие затраты, то такие неэффективные варианты сразу исключаются из рассмотрения.

На второй стадии определим всевозможные сочетания вариантов двух мероприятий и для дальнейших вычислений выберем наиболее экономичные сочетания.

На третьей стадии рассмотрим сочетания лучших вариантов условного высвобождения численности, полученных на второй стадии, с вариантами третьего мероприятия. Снова выберем наиболее эффективные решения. Данные наилучшие решения, получаемые на каждой стадии расчетов, называются **условно оптимальными** решениями, так как они являются оптимальными только для данной конкретной стадии, а не для всего процесса в целом. Эти условно оптимальные решения затем будут использованы для нахождения оптимального результата, и т.д.

На каждой стадии в учет принимаются только такие сочетания вариантов, которые соответствуют дискретным значениям величин условного высвобождения численности работающих, обеспечиваемых двумя, тремя мероприятиями и т.д. Множество дискретных значений необходимо рассматривать по той причине, что на каждой промежуточной стадии еще не известно, какой из вариантов каждого мероприятия будет оптимальным с точки зрения окончательного решения.

На последней стадии выбирается один, наилучший, вариант, обеспечивающий заданное количество высвобождаемых работников с минимальными затратами. Он определяется обратным ходом по условно оптимальным решениям. Общая схема решения задачи может быть представлена следующим образом:



На приведенной схеме $\triangle$ соответствует варианту мероприятия, а $\bigcirc$ – условно оптимальному решению. Кружки и треугольники, выделенные более темным цветом, символизируют оптимальные решения. Обратный ход процесса показан возвратными стрелками.

Рекуррентное соотношение, которое будет использовано при решении задачи, выглядит следующим образом:

$$f_N(B) = \min\{g_N(b_N) + f_{N-1}(B - b_N)\},$$

где N – номер стадии процесса решения задачи;

B – суммарное количество высвобождаемых работников на стадии N , чел.;

b_N – условное высвобождение численности работников, соответствующее варианту мероприятия на стадии N , чел.;

$f_N(B)$ – минимальное значение функции затрат на общее количество высвобождаемых на стадии N , тыс. грн.;

$g_N(b_N)$ – затраты, соответствующие условному высвобождению b_N работников по варианту мероприятия, включаемого в решение на стадии N , тыс. грн.;

$f_{N-1}(B - b_N)$ минимальное значение функции затрат, соответствующее высвобождению $(B - b_N)$ человек на стадии $(N - 1)$.

На первой стадии затраты определяются из выражения

$$f_1(B) = g_1(b_1).$$

Применим метод динамического программирования к решению нашей задачи.

1 стадия.

На данной стадии находим лучшие варианты высвобождения работающих по первому мероприятию. Так,

$f_1(2) = 8$ – лучший вариант высвобождения двух человек работающих, а

$f_1(3) = 11$ – лучший вариант высвобождения трех человек.

2 стадия.

Находим сочетания вариантов двух мероприятий (1-го и 2-го), выбираем условно оптимальные решения

$$f_2(7) = g_2(5) + f_1(7 - 5) = 14 + 8 = 22,$$

$$f_2(8) = g_2(5) + f_1(8 - 5) = 14 + 11 = 25,$$

$$f_2(9) = g_2(7) + f_1(9 - 7) = 18 + 8 = 26,$$

$$f_2(10) = g_2(7) + f_1(10 - 7) = 18 + 11 = 29,$$

3 стадия.

Находим сочетания вариантов третьего мероприятия и условно оптимальных решений, полученных на второй стадии. Выбираем новые условно оптимальные решения (подчеркнуты), соответствующие данной стадии расчетов:

$$f_3(12) = g_3(5) + f_2(12 - 5) = 10 + 22 = 32.$$

$$f_3(13) = \min \left\{ \begin{array}{l} g_3(5) + f_2(13 - 5) \\ \underline{g_3(6) + f_2(13 - 6)} \end{array} \right\} = \min \left\{ \begin{array}{l} 11 + 25 \\ \underline{13 + 22} \end{array} \right\} = 35,$$

$$f_3(14) = \min \left\{ \begin{array}{l} g_3(5) + f_2(14-5) \\ g_3(6) + f_2(14-6) \\ g_3(7) + f_2(14-7) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+26 \\ 13+25 \\ 17+22 \end{array} \right\} = 37,$$

$$f_3(15) = \min \left\{ \begin{array}{l} g_3(5) + f_2(15-5) \\ g_3(6) + f_2(15-6) \\ g_3(7) + f_2(15-7) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+29 \\ 13+26 \\ 17+25 \end{array} \right\} = 39,$$

$$f_3(16) = \min \left\{ \begin{array}{l} g_3(6) + f_2(16-6) \\ g_3(7) + f_2(16-7) \end{array} \right\} = \min \left\{ \begin{array}{l} 13+29 \\ 17+26 \end{array} \right\} = 42,$$

$$f_3(17) = g_3(7) + f_2(17-7) = 17 + 29 = 46.$$

4 стадия.

Находим сочетания вариантов четвертого мероприятия и условно оптимальных решений, полученных на третьей стадии. Выбираем новые условно оптимальные решения

$$f_4(20) = g_4(8) + f_3(20-8) = 11 + 32 = 43,$$

$$f_4(21) = \min \left\{ \begin{array}{l} g_4(8) + f_3(21-8) \\ g_4(9) + f_3(21-9) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+35 \\ 15+32 \end{array} \right\} = 46,$$

$$f_4(22) = \min \left\{ \begin{array}{l} g_4(8) + f_3(22-8) \\ g_4(9) + f_3(22-9) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+37 \\ 15+35 \end{array} \right\} = 48,$$

$$f_4(23) = \min \left\{ \begin{array}{l} g_4(8) + f_3(23-8) \\ g_4(9) + f_3(23-9) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+39 \\ 15+37 \end{array} \right\} = 50,$$

$$f_4(24) = \min \left\{ \begin{array}{l} g_4(8) + f_3(24-8) \\ g_4(9) + f_3(24-9) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+42 \\ 15+39 \end{array} \right\} = 53,$$

$$f_4(25) = \min \left\{ \begin{array}{l} g_4(8) + f_3(25-8) \\ g_4(9) + f_3(25-9) \end{array} \right\} = \min \left\{ \begin{array}{l} 11+46 \\ 15+42 \end{array} \right\} = 57,$$

$$f_4(26) = g_4(9) + f_3(24-9) = 15 + 46 = 61.$$

5 стадия.

На данной стадии рассматривается только один вариант сочетаний мероприятий, приносящий условное высвобождение численности работников в количестве 33 человек:

$$f_5(33) = \min \left\{ \begin{array}{l} g_5(8) + f_4(33-8) \\ g_5(10) + f_4(33-10) \\ g_5(11) + f_4(33-11) \end{array} \right\} = \min \left\{ \begin{array}{l} 18 + 57 \\ 24 + 50 \\ 30 + 48 \end{array} \right\} = 74.$$

Таким образом, минимальные затраты, обеспечивающие условное высвобождение 33 человек, составляют 74 тыс. грн.

Оптимальное сочетание мероприятий находится обратным ходом по условно оптимальным решениям. Результаты расчетов сведены в таблицу. Оптимальные варианты мероприятий выделены.

Вариант	Мероприятия									
	1		2		3		4		5	
	В	К	В	К	В	К	В	К	В	К
1	2	8	5	14	5	11	8	11	8	18
2	3	11	7	18	6	13	9	15	10	24
3					7	17			11	30

Решение задачи методом динамического программирования удобно производить в таблицах. Так, на первой стадии будут рассмотрены только первое и второе мероприятия. Показатели условного высвобождения работников и соответствующие им затраты представлены в заголовках таблицы, сочетания вариантов условного высвобождения (и затраты) – в клетках на пересечении

1 меропр.	2	3
2 меропр.	8	11
5	7	8
14	22	25
7	9	10
18	26	29

Здесь и далее выделенные клетки соответствуют условно оптимальным решениям. Присоединяем к лучшим сочетаниям, полученным на данной стадии, варианты третьего мероприятия:

1+2 меропр.	7	8	9	10
3 меропр.	22	25	26	29
5	12	13	14	15
11	33	36	37	40
6	13	14	15	16
13	35	38	39	42
7	14	15	16	17
17	39	42	43	46

Добавляем варианты четвертого мероприятия. Получаем новую таблицу

1+2+3 меропр. 4 меропр.	12 33	13 35	14 37	15 39	16 42	17 46
8 11	20 44	21 46	22 48	23 50	24 53	25 57
9 15	21 48	22 50	23 52	24 55	25 57	26 61

На последней стадии также формируется таблица, однако в нее нет смысла включать все условно оптимальные решения, полученные на предыдущей стадии расчетов. Возьмем только те из них, которые в сочетании с вариантами пятого мероприятия дадут условное высвобождение, равное 33 работникам:

1+2+3+4 меропр. 5 меропр.	...	22 48	23 50	...	25 57	...
8 18					33 75	
10 24			33 74			
11 30		33 78				

Таким образом, получен тот же результат, что и при использовании рекуррентного соотношения Р.Беллмана.

Поиск оптимальных сочетаний вариантов мероприятий осуществляется обратным ходом по таблицам. Овалами и стрелками показаны взаимосвязи между соседними стадиями расчетов при нахождении оптимального решения.

1+2+3 меропр. 4 меропр.	12 33	13 35	14 37	15 39	16 42	17 46
8 11	20 44	21 46	22 48	23 50	24 53	25 57
9 15	21 48	22 50	23 52	24 55	25 57	26 61

1+2 меропр. 3 меропр.	7 22	8 25	9 26	10 29
5 11	12 33	13 36	14 37	15 40
6 13	13 35	14 38	15 39	16 42
7 17	14 39	15 42	16 43	17 46

1 меропр. 2 меропр.	2 8	3 11
5 14	7 22	8 25
7 18	9 26	10 29

Рассмотренные подходы к решению задач методом динамического программирования не являются единственными. Существуют и другие, которые учитывают нелинейный характер рекуррентных соотношений. Особым разделом являются вероятностные задачи, которые решаются методом динамического программирования.

ЗАДАЧИ

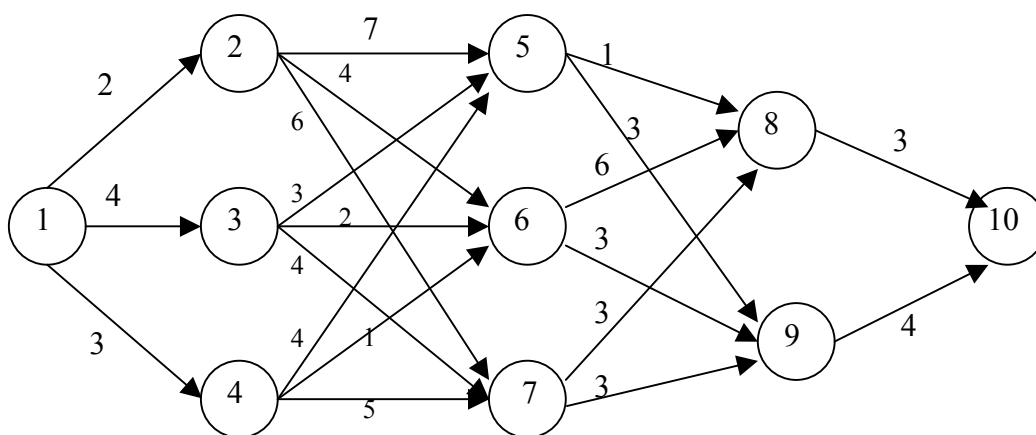
- 9.1. Торговый агент живет в Днепропетровске. Сегодня воскресенье, а в следующий четверг он должен быть в Харькове. В каждый из дней (понедельник, вторник, среду) он может продавать свои товары в Харькове, Днепропетровске и Киеве. Последний опыт показывает, что день в Харькове принесет ему 22 грн. дохода, день в Днепропетровске – 26 грн. и день в Киеве – 27 грн.

Затраты на проезд приведены в таблице

	Харьков	Днепропетровск	Киев
Харьков	–	15	12
Днепропетровск	15	–	17
Киев	12	17	–

Где должен провести первые три дня недели торговый агент, чтобы получить наибольшую прибыль (доход минус транспортные расходы)?

- 9.2. Найдите на сети кратчайший путь от вершины 1 до вершины 10. Найдите также кратчайший путь от вершины 3 до вершины 10.

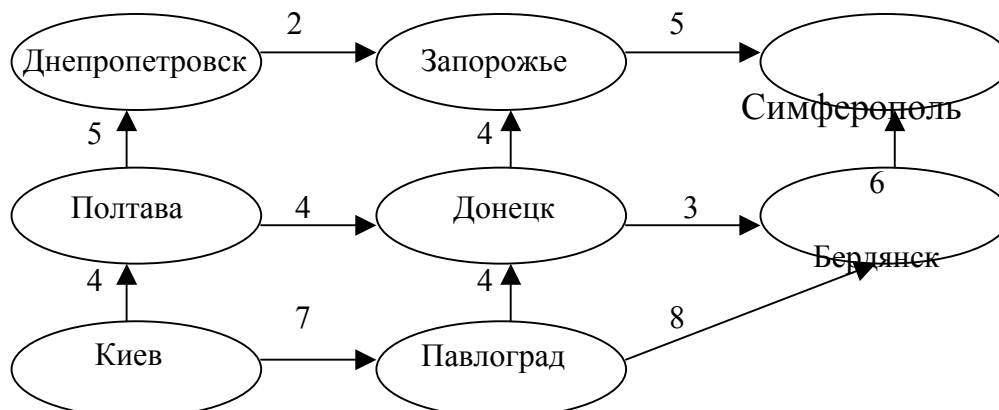


- 9.3. Кассир магазина канцелярских товаров должен дать сдачу клиенту с 2-х грн. за покупку материалов на сумму 1 грн. 09 коп. Желание кассира – дать сдачу как можно меньшим количеством монет.

Используйте динамическое программирование чтобы определить, каким набором монет клиент получит причитающуюся ему сдачу.

Как изменится ответ, если в распоряжении кассира в дополнение к циркулирующим в Украине будут монеты достоинством 20 копеек?

- 9.4. Мне необходимо проехать от Киева до Симферополя. Возможны несколько путей (см. сеть). Число, соответствующее каждой дуге – количество часов, которое необходимо затратить для поездки между парой городов.



Двигаясь в обратном направлении, найдите кратчайший путь (во временных единицах) от Киева до Симферополя.

(Подсказка: двигаясь назад, не думайте о стадиях, а только о состояниях).

- 9.5. Компания «Авиалинии Донбасса» должна определить недельное расписание полетов из Донецка в другие города Украины. Между каждой парой городов может быть установлено до 6 рейсов в сутки. Пунктами назначения могут быть Киев, Одесса и Львов. В таблице приведен размер прибыли компании в зависимости от места назначения и частоты рейсов.

Пункт назначения	Количество рейсов					
	1	2	3	4	5	6
Львов	80	150	210	250	270	280
Одесса	100	195	275	325	300	250
Киев	90	180	265	310	350	320

Определите оптимальное количество полетов между Донецком и остальными городами, обеспечивающее наибольшую прибыль.

Как должен измениться ответ, если число возможных рейсов сократится до 4-х?

- 9.6. Предположим, что новая машина стоит 20000 грн. Затраты на обслуживание (за год) и стоимость ее перепродажи представлены в таблице

Возраст машины, лет.	Цена перепродажи, грн.	Затраты на обслуживание, грн.
1	14000	600 (год 1)
2	12000	1000 (год 2)
3	8000	1600 (год 3)
4	6000	2400 (год 4)
5	4000	3200 (год 5)
6	2000	4400 (год 6)

Определите оптимальную политику замены машины, которая минимизирует затраты на ее владение и содержание на шесть лет вперед при условии, что машина только что приобретена.

- 9.7. На предстоящие шесть лет компании необходима работающая машина. В настоящее время она имеет новую машину. В начале каждого нового года компания может либо оставить прежнюю машину, либо продать ее и купить новую. Машина не может эксплуатироваться более трех лет. Новая машина стоит 5000 грн. Доход от ее эксплуатации, затраты на содержание в течение года и остаточная стоимость к концу года приведены в таблице

Показатели	Возраст машины в начале года		
	0 лет	1 год	2 года
Доход, грн.	4500	3000	1500
Затраты на обслуживание, грн.	500	700	1100
Остаточная стоимость, грн.	3000	1800	500

Используйте динамическое программирование для определения максимальной прибыли, получаемой компанией в течение последующих 6-ти лет.

- 9.8. Компании требуется следующее количество работников на последующие 5 лет (по годам): 1 год – 15; 2 год – 30; 3 год – 10; 4 год – 30, 5 год – 20. В настоящее время в компании работают 20 человек. Каждый работник получает 3000 грн. в год. В начале каждого года работник может быть нанят на работу или уволен. Затраты, связанные с наймом одного работника, составляют 1000 грн., а с его увольнением – 2000 грн. Вновь нанятый работник может быть использован для удовлетворения потребности компании в текущем году. В течение каждого года 10% работников уходят со службы по собственному желанию (работники, покидающие компанию по собственному желанию, не требуют затрат на увольнение).

Сформулируйте рекуррентное соотношение, которое могло бы быть использовано для минимизации общих затрат, связанных с удовлетворением потребности в работниках на предстоящие 5 лет.

Как должно быть модифицировано рекуррентное соотношение, если нанятые работники смогут быть использованы для удовлетворения потребности в рабочей силе только в следующем году.

- 9.9. Используйте динамическое программирование для решения задачи о ранце, описанной следующей математической моделью:

$$\begin{aligned} \text{Max } Z &= 5x_1 + 4x_2 + 2x_3 \\ 4x_1 + 3x_2 + 2x_3 &\leq 8 \\ x_1, x_2, x_3 &\geq 0 - \text{целые.} \end{aligned}$$

- 9.10. На промышленном предприятии намечается внедрить систему мер по повышению эффективности производства и росту производительности труда. В настоящее время на предприятии работает 1000 человек. Предполагается,

что в новом году объем выпуска продукции возрастет на 2.5%. Данный прирост намечается обеспечить за счет роста производительности труда на 4.5%. Для достижения заданных показателей деятельности предприятия разработана система мероприятий. Причем по каждому направлению их внедрения имеются альтернативные варианты, отличающиеся показателями затрат и условного высвобождения численности работников (как показателя, отражающего рост производительности труда). Указанные мероприятия приведены в таблице

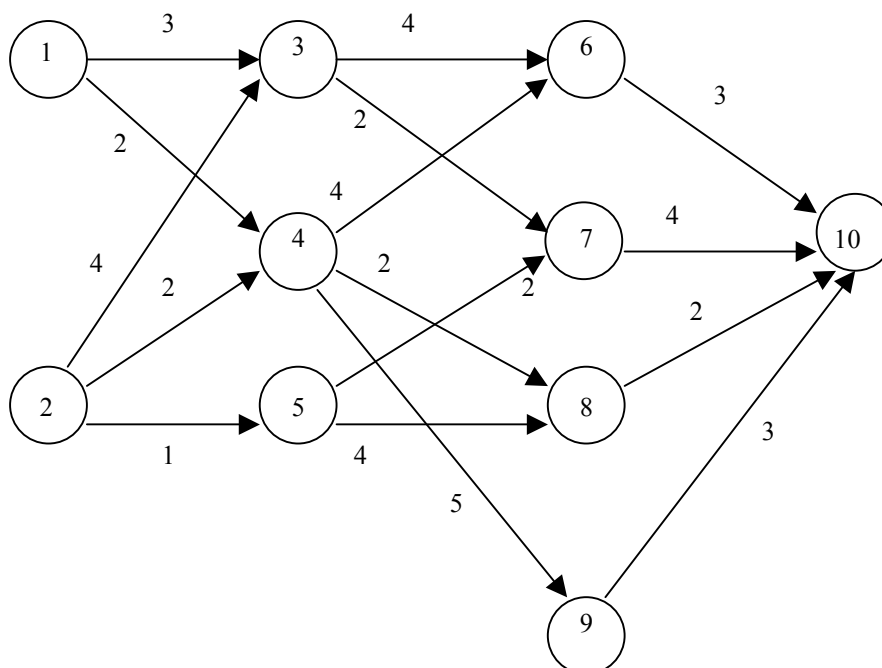
Вариант	Мероприятия									
	1		2		3		4		5	
	В	К	В	К	В	К	В	К	В	К
1	7	18	6	14	6	15	8	10	8	28
2	8	21	7	18	8	18	10	14	10	32
3	9	26	8	23	10	22	11	17	11	44

В таблице использованы следующие обозначения: В – условное высвобождение численности работающих, чел.; К – затраты на реализацию мероприятия, тыс. грн.

Необходимо определить оптимальное сочетание мероприятий, обеспечивающее достижение заданных показателей с минимальными затратами.

Является ли полученное решение единственным?

- 9.11. Для приведенной сети найдите кратчайшее расстояние от вершины 1 до вершины 10 и от вершины 2 до вершины 10.



- 9.12. Компания должна удовлетворить следующие требования в комплектующих изделиях на последующие 4 месяца: первый месяц – 1 единица; второй месяц – 1 единица; третий месяц – 2 единицы; четвертый месяц – 2 единицы. Стоимость оформления заказа – 4 грн., затраты, связанные с хранением единицы комплектующих изделий на конец месяца, –

2 грн. В начале первого месяца 1 единица комплектующих изделий имеется в распоряжении компании. Заказы доставляются мгновенно.

Используйте процедуру движения от конца к началу процесса для определения оптимальной политики по заказу комплектующих изделий.

Используйте процедуру движения от начала к концу процесса для определения оптимальной политики по заказу комплектующих изделий.