

МЕТОДЫ ОПТИМИЗАЦИИ

**3. МЕТОДЫ РЕШЕНИЯ
КОМБИНАТОРНО-
ОПТИМИЗАЦИОННЫХ ЗАДАЧ**

3.1. СТРАТЕГИИ ДЕКОМПОЗИЦИИ МНОЖЕСТВА РЕШЕНИЙ И ДЕРЕВО ПОИСКА

Широкий круг методов поиска решения комбинаторно-оптимизационных задач основан на двух идеях: декомпозиция пространства решений и выбор перспективного подмножества вариантов на основе некоторой оценки.

Рассмотрим процесс декомпозиции. Множество возможных вариантов решения M разбивается в соответствии с каким-то принципом, как правило, на непересекающиеся подмножества M такие, что $\cup M_i = M$. Далее, используя тот же принцип, полученные подмножества вновь разбиваются на подмножества, включающие меньшее количество вариантов. После некоторого шага разбиения каждое множество содержит по одному варианту решения. Сопоставим каждому подмножеству вершину графа. Соединив ребром вершины, соответствующие подмножествам M_i и M_j (если подмножество M_j получено непосредственным разбиением подмножества M_i), придем к так называемому *дереву решений*.

Принцип разбиения пространства решений связан с видом преобразований, которые необходимо выполнить над графом для получения графа результата.

Проиллюстрируем сказанное примером поиска простых цепей (маршрутов) из вершины x_1 в вершину x_4 в графе $G(X, U)$, показанном на рис. 3.1, а. Начиная с вершины x_1 будем включать в маршруты ребра *в порядке возрастания их номеров*. Вершина 0 дерева решений (рис. 3.1, б) соответствует множеству всех возможных вариантов маршрутов из x_1 в x_4 . Принцип разбиения этого множества в данном случае естественно вытекает из процедуры последовательно-параллельного прослеживания вариантов маршрутов по ребрам графа G , т.е. последовательного включения ребер в возможные маршруты.



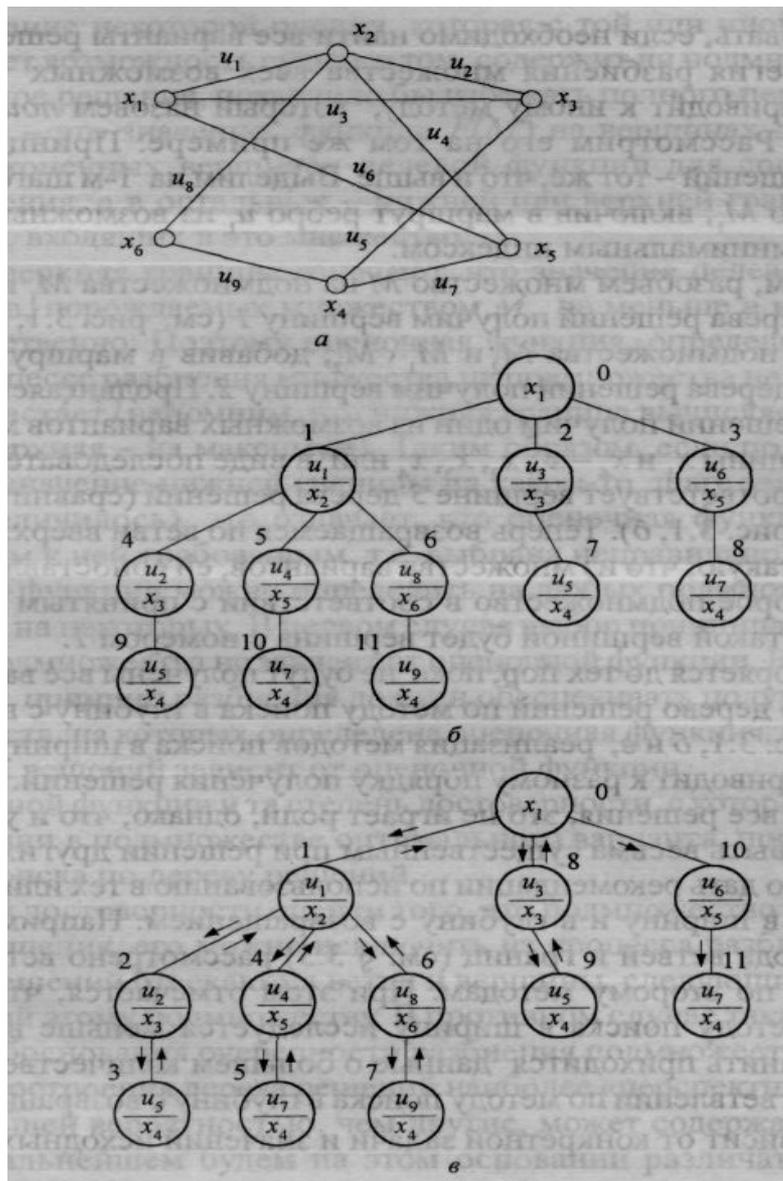


Рис. 3.1. Граф $G(X, U)$ (а) и деревья решений поиска простых цепей по методу в ширину (б) и в глубину с возвращением (в)



Вершина дерева решений с номером 1 соответствует включению в маршрут ребра u_1 вершины с номерами 2 и 3 - ребер u_3 и u_6 . Таким образом, на первом шаге (на 2-м уровне дерева решений) множество M разбивается на три непересекающихся подмножества вариантов, одно из которых M_1 , соответствующее вершине 1 дерева решений, порождает следующие три варианта простых цепей графа G : $\{u_1, u_2, u_5\}$, $\{u_1, u_4, u_7\}$ и $\{u_1, u_8, u_9\}$. Далее процесс очевиден. Обратим внимание на то, что в данном случае на следующий уровень дерева решений переходим только после того, как были получены все подмножества текущего уровня.

Описанный процесс реализует получение всех вариантов решения по *методу поиска в ширину* с последовательным формированием состава вариантов и соответствует полному перебору. Очевидно, что в рассмотренном виде этот метод можно использовать, если необходимо найти все варианты решения.

Другая стратегия разбиения множества всех возможных вариантов на подмножества приводит к иному методу, который назовем *поиск в глубину с возвратом*. Рассмотрим его на том же примере. Принцип разбиения пространства решений - тот же, что и выше. Выделим на 1-м шаге из множества M подмножество M_1 включив в маршрут ребро u_1 из возможных $\{u_1, u_3, u_6\}$ в соответствии с минимальным индексом.

Таким образом, разобьем множество M на подмножества M_1 и $M \setminus M_1$. Тогда на 2-м уровне дерева решений получим вершину 1 (см. рис. 3.1, в). Множество M_1 разобьем на подмножества M_2 и $M_1 \setminus M_2$, добавив в маршрут ребро u_2 . На третьем уровне дерева решений получим вершину 2. Продвигаясь по этой ветви в глубь дерева решений получим один из возможных вариантов маршрута, соединяющего вершины x_1 и x_4 - x_1, x_2, x_3, x_4 или в виде последовательности ребер - u_1, u_2, u_5 , что соответствует вершине 3 дерева решений (сравните номер такой же вершины на рис. 3.1, б). Теперь возвращаемся по ветви вверх, пока не придем в вершину, такую, что из множества вариантов, ей сопоставленных, можно выделить некоторое подмножество в соответствии с принятым принципом. В нашем примере такой вершиной будет вершина с номером 1.



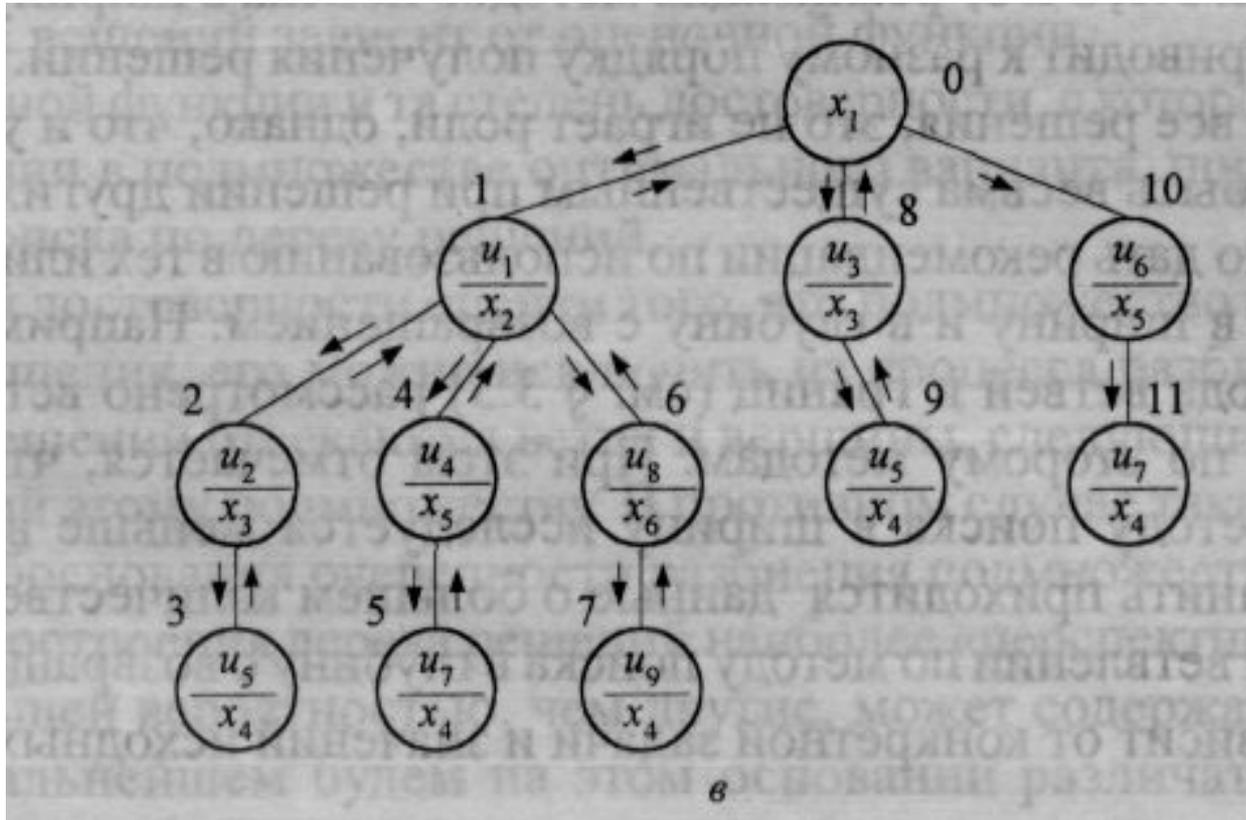
Вершина дерева решений с номером 1 соответствует включению в маршрут ребра u_1 вершины с номерами 2 и 3 - ребер u_3 и u_6 . Таким образом, на первом шаге (на 2-м уровне дерева решений) множество M разбивается на три непересекающихся подмножества вариантов, одно из которых M_1 , соответствующее вершине 1 дерева решений, порождает следующие три варианта простых цепей графа G : $\{u_1, u_2, u_5\}$, $\{u_1, u_4, u_7\}$ и $\{u_1, u_8, u_9\}$. Далее процесс очевиден. Обратим внимание на то, что в данном случае на следующий уровень дерева решений переходим только после того, как были получены все подмножества текущего уровня.

Описанный процесс реализует получение всех вариантов решения по *методу поиска в ширину* с последовательным формированием состава вариантов и соответствует полному перебору. Очевидно, что в рассмотренном виде этот метод можно использовать, если необходимо найти все варианты решения.

Другая стратегия разбиения множества всех возможных вариантов на подмножества приводит к иному методу, который назовем *поиск в глубину с возвратом*. Рассмотрим его на том же примере. Принцип разбиения пространства решений - тот же, что и выше. Выделим на 1-м шаге из множества M подмножество M_1 включив в маршрут ребро u_1 из возможных $\{u_1, u_3, u_6\}$ в соответствии с минимальным индексом.

Таким образом, разобьем множество M на подмножества M_1 и $M \setminus M_1$. Тогда на 2-м уровне дерева решений получим вершину 1 (см. рис. 3.1, в). Множество M_1 разобьем на подмножества M_2 и $M_1 \setminus M_2$, добавив в маршрут ребро u_2 . На третьем уровне дерева решений получим вершину 2. Продвигаясь по этой ветви в глубь дерева решений получим один из возможных вариантов маршрута, соединяющего вершины x_1 и x_4 - x_1, x_2, x_3, x_4 или в виде последовательности ребер - u_1, u_2, u_5 , что соответствует вершине 3 дерева решений (сравните номер такой же вершины на рис. 3.1, б). Теперь возвращаемся по ветви вверх, пока не придем в вершину, такую, что из множества вариантов, ей сопоставленных, можно выделить некоторое подмножество в соответствии с принятым принципом. В нашем примере такой вершиной будет вершина с номером 1.





Процесс повторяется до тех пор, пока не будут получены все варианты маршрута. Построено дерево решений по методу поиска в глубину с возвратением. Как видно из рис. 3.1. (б и в), реализация методов поиска в ширину и в глубину с возвратением приводит к разному порядку получения решений. В данном случае, когда ищем все решения, это не играет роли, однако, что и увидим в дальнейшем, может быть весьма существенным при решении других задач.

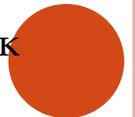


3.2. МЕТОДЫ ПОИСКА РЕШЕНИЯ, ИСПОЛЬЗУЮЩИЕ ИДЕЮ ОТСЕЧЕНИЯ

В практике проектирования задачи нахождения всех возможных решений довольно редки. Обычно требуется найти одно решение, наилучшее в смысле некоторого функционала - целевой функции. Поиск такого решения декомпозицией множества вариантов по методу в ширину или в глубину с возвращением подразумевает вычисление значений целевой функции для каждого варианта, сравнение их и выбор оптимального, т.е. приводит к полному перебору. Полный перебор требует слишком много времени либо просто нереализуем по этой причине при достаточно большой размерности задачи.

Существование некоторой оценки, которая с той или иной степенью достоверности дает возможность судить о том, содержит ли подмножество вариантов оптимальное решение, позволило бы избежать полного перебора. В общем случае оценка - это значение функции $F(M_i)$ на вершинах дерева решений, равная в его конечных вершинах целевой функции для соответствующего варианта решения, а в остальных - нижней или верхней границе функции F для вариантов, входящих в это множество.

Нижняя и верхняя границы означают, что значение целевой функции для всех вариантов, порождаемых множеством M_i , не меньше и не больше числа $F(M_i)$ соответственно. Поэтому оценочная функция, определяющая нижнюю границу, в процессе разбиения множества на подмножества не убывает, а верхнюю - не возрастает (напомним, что нижняя граница вычисляется в задачах на минимум, а верхняя - на максимум). Таким образом, если по мере разбиения множества M значение нижней границы на каком-то шаге уменьшилось (для верхней - увеличилось), это означает, что оценочная функция не отвечает предъявляемым к ней требованиям, т.е. выбрана неправильно.



Оценочные функции можно определить на любых подмножествах вариантов либо лишь на некоторых. В первом случае выбор принципа разбиения множества M на подмножества не зависит от оценочной функции. Во втором случае используемый принцип разбиения должен обеспечивать получение только таких подмножеств, на которых определена оценочная функция. Отсюда следует, что вид дерева решений зависит от оценочной функции.

Вид оценочной функции и та степень достоверности, с которой можно судить по ней о наличии в подмножестве оптимального варианта, порождает различные методы поиска по дереву решений.

При полной достоверности оценки того, что подмножество не содержит оптимального решения, его можно исключить из процесса разбиения - говорят, что в дереве решений отсекаются ветви и вершины, следующие за вершиной, сопоставленной этому подмножеству. В противном случае такая оценка может служить для обоснования очередности разбиения подмножеств, т.е. выбора на каждом шаге построения дерева решений наиболее «перспективной» вершины, которая с большей вероятностью, чем другие, может содержать оптимальное решение. В дальнейшем будем на этом основании различать *отсекающую оценку и оценку перспективности*.

Выбор оценочной функции - один из самых сложных и ответственных этапов разработки алгоритмов, использующих методы поиска с отсечением. От него зависит как эффективность таких алгоритмов, так и их точность. В свою очередь возможности оценочной функции в значительной степени определяются спецификой задачи и математическими свойствами графов - объекта и результата проектирования.

Сделаем несколько общих замечаний. Оценочная функция в виде верхней или нижней границы целевой функции (возможно ее текущего значения, например сумма длин ребер построенного участка маршрута) обычно является основанием для выбора более перспективного подмножества, но может служить и отсекающей оценкой. В качестве отсекающей оценки для большинства комбинаторно-оптимизационных задач может выступать опорное решение, т.е. значение целевой функции какого-либо варианта решения. Несомненно, что если для задачи на минимум для какого-то подмножества нижняя граница или текущее значение целевой функции больше или равно значению целевой функции некоторого решения, это подмножество не содержит оптимального варианта. Однако трудно ожидать выполнения этого условия на ранних этапах разбиения пространства решений.

Метод поиска в глубину. Этот метод является вырожденным случаем разбиения пространства решений. Он заключается в том, что на каждом уровне дерева решений в соответствии с некоторой оценкой, определяемой спецификой задачи, выбирают одно подмножество, не формируя (отсекая) остальные, до тех пор, пока не получим решение. Таким образом, здесь формируется только одна ветвь дерева решений. Очевидно, что алгоритмы, основанные на этом методе, должны быть весьма эффективными в смысле затрат времени и памяти ЭВМ. Если оценка выбора является отсекающей, т.е. с вероятностью, равной единице, позволяет судить о том, что оптимальный вариант находится в данном подмноестве, получаем точное решение. В противном случае гарантируется лишь приближенное. Это положение имеет очень большое значение, так как в большинстве источников утверждается, что последовательные алгоритмы, реализующие метод поиска в глубину, гарантируют лишь приближенное решение.

Проиллюстрируем сказанное примером идеи алгоритма Прима, предназначенного для построения остовного дерева минимального веса графа G . В § 4.1 в ходе доказательства правильности этого алгоритма будет показано, что выбор на каждом шаге ребра минимальной длины (не образующего цикл) обеспечивает получение точного решения. Таким образом, минимальная длина очередного ребра является достоверной оценкой, а включение в строящееся дерево такого ребра отсекает все вершины дерева решений, которые соответствуют вариантам остовного дерева графа G , не содержащим этого ребра. На рис. 3.2 показаны полный четырех вершинный граф G , дерево решений и полученное остовное дерево минимальной длины; здесь M_1 и M_2 – все варианты деревьев графа $G(X, U)$, содержащих и не содержащих ребро u_1 ; $M_2 \subset M_1$ – все варианты деревьев, включающих ребра u_1 и u_5 ; и $M_3 \subset M_1$ – все варианты деревьев, содержащих ребро u_1 и не содержащих ребро u_5 ; M_3 – остовное дерево минимальной длины графа $G: \{u_1, u_5, u_6\}$, вес которого равен 12.



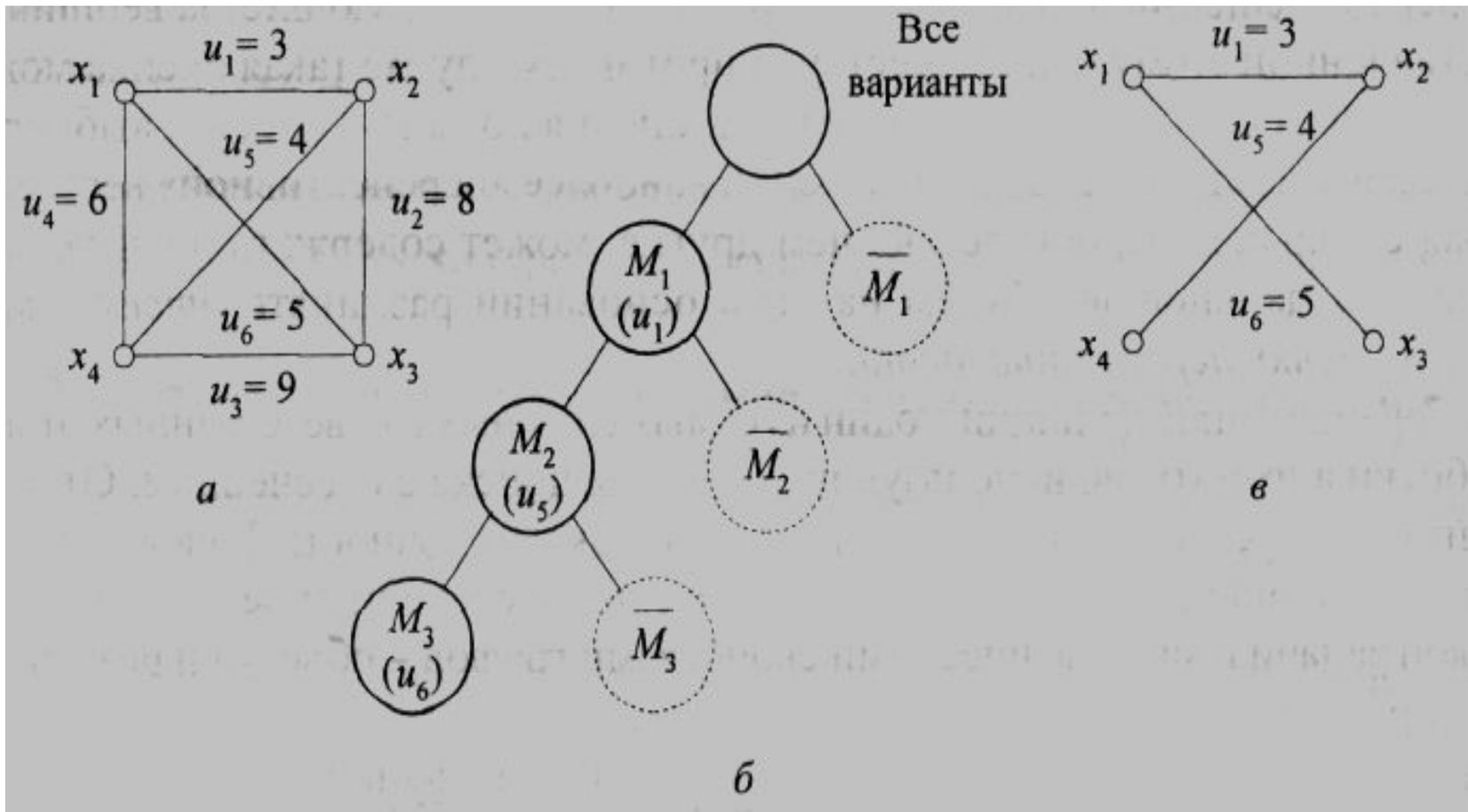


Рис. 3.2. Полный граф $G(X, U)$ (а), дерево решений (б) и остовное дерево минимальной длины (в)



Метод параллельного поиска. Метод параллельного или одновременного поиска относится к классу эвристических и представляет упрощенный вариант метода поиска в ширину. Он заключается в следующем: множество решений разбивается на подмножества 1-го уровня, каждая вершина которого становится началом ветви, формируемой по методу поиска в глубину. Переход к формированию подмножеств следующего уровня дерева решений выполняется после получения подмножеств всех ветвей текущего уровня. Дерево решений, иллюстрирующее идею метода, показано на рис. 3.3; здесь n - количество искомых решений, следовательно, на j -м уровне дерева n подмножеств $M_i^j, i = \overline{1, n}$ - множество индексов этих подмножеств (верхний индекс соответствует номеру уровня дерева решений).

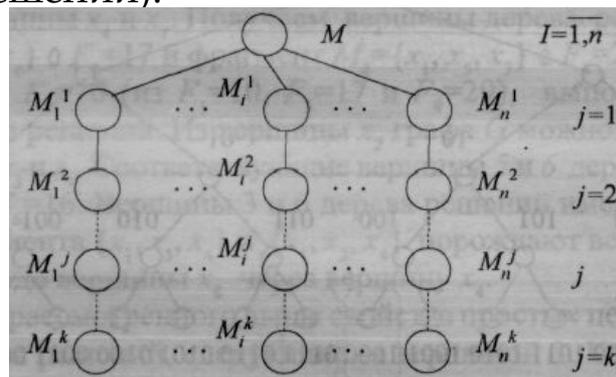


Рис. 3.3. Дерево решений по методу параллельного поиска

В зависимости от специфики задачи подмножества разных ветвей могут быть как пересекающимися, так и непересекающимися. Рассмотрим задачу компоновки схемы в n конструктивных модулей. Множество элементов схемы \mathcal{E} поставлено во взаимно-однозначное соответствие множеству вершин X гиперграфа $\mathcal{E} \leftrightarrow X$. Указанная задача может быть решена алгоритмом, реализующим метод параллельного поиска (алгоритм параллельного разрезания гиперграфа см. § 5.3). Результатом его работы будет разбиение множества вершин гиперграфа на n подмножеств $X_i, i = \overline{1, n}$. Так как один и тот же элемент схемы не может входить в разные конструктивные модули и состав X_i определяется по методу поиска в глубину последовательным включением вершин $x \in X$ в X_i^1 (X_i^1 соответствует M_i^1), то $X_i^j \cap X_p^j = \emptyset$ для всех $i, p \in I = \overline{1, n}$. Таким образом, в данной задаче подмножества, принадлежащие разным ветвям дерева решений, должны быть непересекающимися.

Если подмножества вариантов 1-го уровня содержат все варианты решения, т.е. удовлетворяют условию $\cup M_i^1 = M$ и оценка выбора подмножества в каждой ветви является отсекающей, то метод обеспечивает точное решение.

Вырожденным случаем оценочной функции является невычисляемая отсекающая оценка, например, некоторое заранее сформулированное условие. Примером может служить задача о кодовом замке. Пусть кодовый замок имеет четыре разряда, и каждый разряд может принимать значения «0» и «1». Мы забыли код, но помним, что в нем не может быть подряд трех нулей или единиц - условие отсечения. Мы готовы перепробовать все коды, но хотели бы избежать этого. Нам нужен алгоритм для систематического генерирования четырехразрядных комбинаций из «0» и «1». Учитывая вероятность того, что мы откроем замок раньше, чем будут испытаны все комбинации, целесообразно использовать метод поиска в глубину с возвращением. Примем направление обхода дерева решений слева направо. Принцип разбиения множества решений - присваивание соответствующему разряду значения «0» или «1» (количество сгенерированных разрядов равно номеру уровня дерева решений). Очевидно, что дерево будет бинарным. Тогда алгоритм можно сформулировать так: движемся вниз по дереву, придерживаясь левой ветви, пока не получим полной комбинации либо не выясним нецелесообразность этого (рис. 3.4 - отсечение левой ветви после вершины с кодом 11). Если комбинация не подходит, либо ее или некоторое множество комбинаций нет смысла набирать (отсечение по условию), то возвращаемся в ближайшую вершину и пробуем спускаться по другой ветви (левая ветвь - разряд ставим в положение «1», правая - в «0»). В некоторых источниках [4] этот метод называют «программирование с отходом назад».



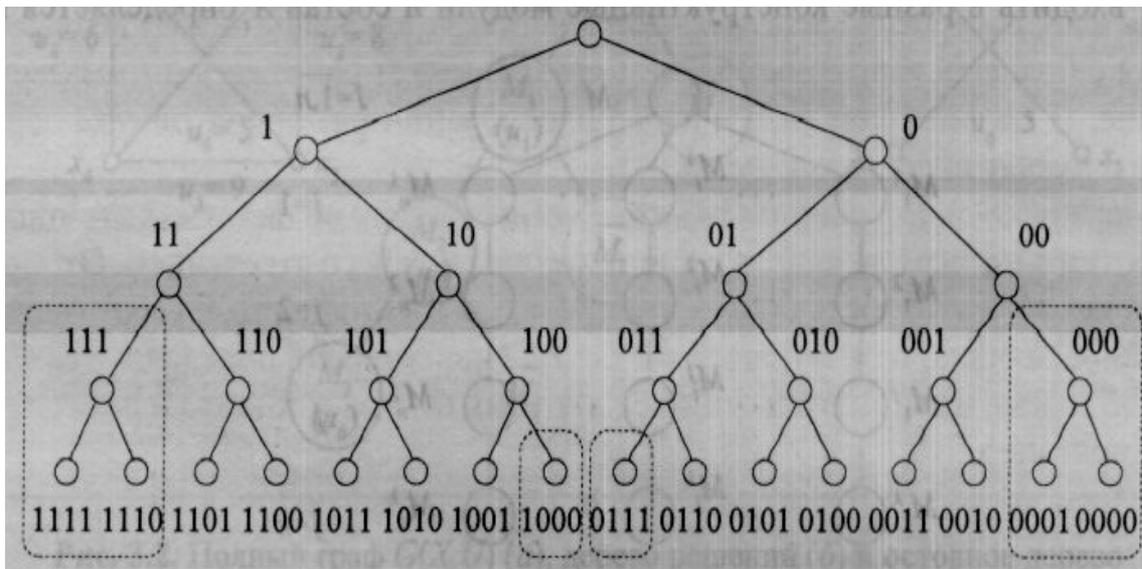


Рис. 3.4. Двоичное дерево поиска кода по методу в глубину с возвращением

Рассмотрим пример использования текущего значения целевой функции в качестве оценки перспективности во всех вершинах дерева решений и этой же величины в качестве отсекающей - в «особых» вершинах дерева решений. В §3.1 коротко рассмотрена задача нахождения всех простых цепей из исходной вершины графа в некоторую конечную. Модифицируем ее: необходимо найти простую цепь (маршрут) из некоторой вершины x_i в вершину x_j графа $G(X, U)$, такой, чтобы суммарная длина ребер, его составляющих, была минимальна.

Примем стратегию формирования дерева решения по методу поиска в ширину, Принцип разбиения множества на подмножества тот же, что и выше, т.е. включение в фрагмент маршрута некоторого ребра. В качестве оценочной функции примем суммарную длину ребер, составляющих фрагмент маршрута. Нетрудно убедиться, что эта величина является нижней границей длины маршрута. Действительно - это функция, определенная на вершинах дерева решений, в каждой конечной вершине она принимает значение целевой функции для соответствующего варианта маршрута и, наконец, функция возрастающая, так как длины ребер - величины положительные. Так как доказательства того, что эта оценка является отсекающей, нет, будем использовать ее в качестве оценки перспективности ветви.



Разные простые цепи, соединяющие вершину x_i с вершиной x_k могут проходить через одну и ту же вершину x_k . Это свойство позволяет в особых точках дерева решений использовать суммарную длину ребер как отсекающую оценку. Отсюда, если суммарная длина ребер от вершины x_i до вершины x_k фрагмента маршрута M_r больше, чем фрагмента маршрута M_t то все варианты маршрутов, порождаемые фрагментом M_r следует отбросить.

На рис. 3.5, а показан граф $G(X, U)$. Пусть исходной является вершина x_1 , а конечной - x_8 . На 1 - м уровне дерева решений (см. рис. 3.5, б) множество возможных вариантов маршрута из вершины x_1 разбивается на два подмножества: маршруты из x_1 через вершину x_2 (вершина дерева решений 1) и через вершину x_5 (вершина дерева решений 2) - длины фрагментов $F_1 = 10$ и $F_2 = 8$ соответственно. На основании оценки перспективности выбираем вторую вершину дерева решений и выполняем ветвление в ней. Из вершины x_5 графа G можно продолжать маршруты в вершины x_4 и x_7 . Получаем вершины дерева решений 3 и 4 (фрагмент $M_3 = \{x_1, x_5, x_4\}$ с $F_3 = 17$ и фрагмент $M_4 = \{x_1, x_5, x_7\}$ с $F_4 = 29$). Выбирая минимальную оценку $F_1 = 10$ (из $F_1 = 10$, $F_3 = 17$ и $F_4 = 29$), выполняем ветвление в вершине 1 дерева решений. Из вершины x_2 графа G можно продолжить маршруты в вершины x_3 и x_4 . Соответствующие вершины 5 и 6 дерева решений имеют оценку $F_5 = 41$ и $F_6 = 16$. Вершины 3 и 6 дерева решений имеют оценки $F_3 = 17$ и $F_6 = 16$ и оба фрагмента $\{x_1, x_5, x_4\}$ и $\{x_1, x_2, x_4\}$ порождают все маршруты от вершины x_1 графа G до вершины x_8 через вершину x_4 .

На основании рассмотренного выше свойства простых цепей отсекаем ребра и вершины дерева решений, следующие за вершиной 3. Этот прием используется в алгоритме Дейкстры [4].



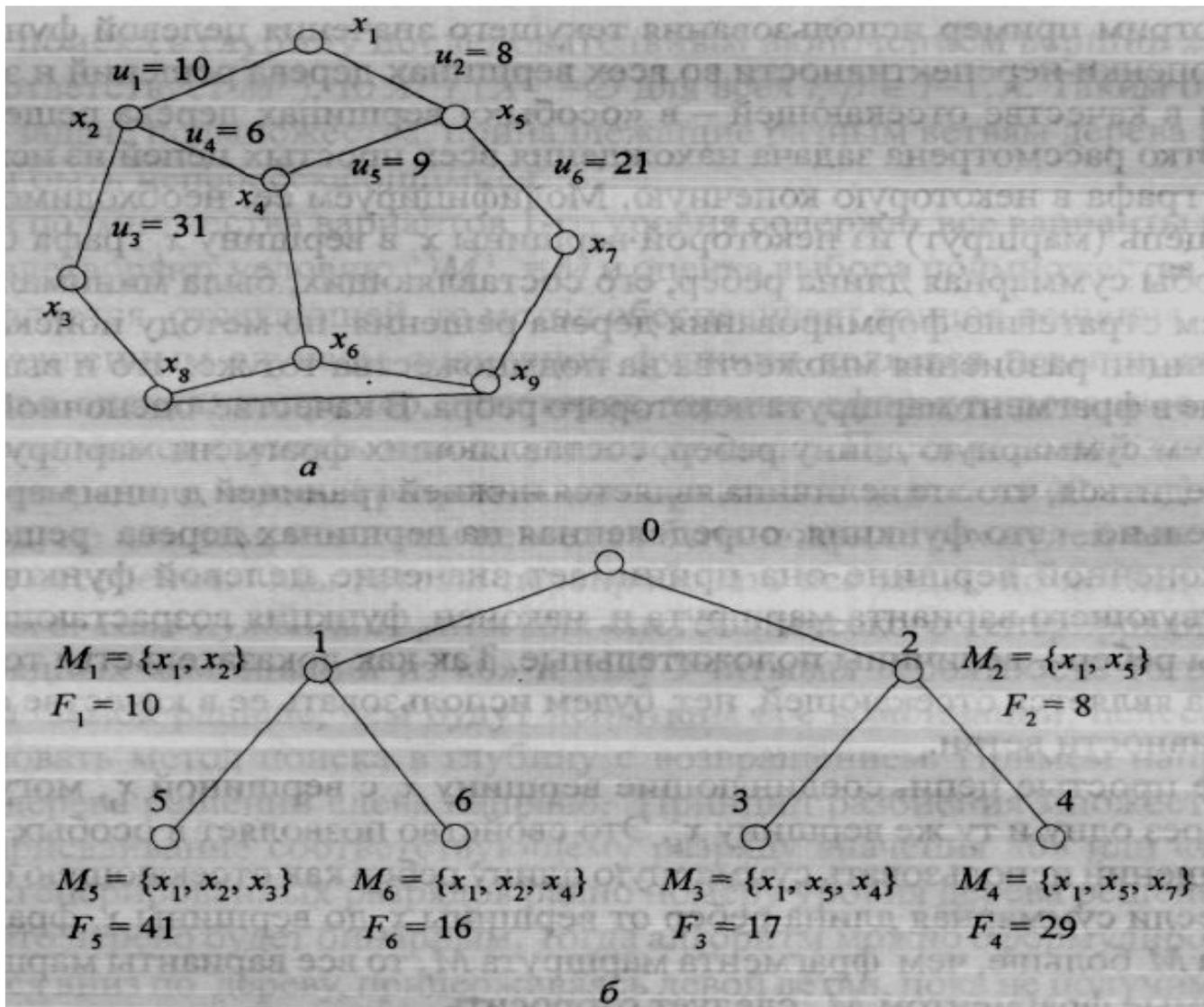


Рис. 3.5. Граф $G(X, U)$ (а) и дерево поиска простой цепи минимальной длины (б)

Изложенный в данном разделе материал позволил подойти к наиболее общему методу решения комбинаторно-оптимизационных задач - методу ветвей и границ. При рассмотрении этого метода коснемся вопроса конструирования оценочной функции для вычисления нижней и верхней границ.



3.3. МЕТОД ВЕТВЕЙ И ГРАНИЦ

Метод ветвей и границ - универсальный, хотя и достаточно сложный метод точного решения широкого круга комбинаторно-оптимизационных задач направленным перебором вариантов при условии, что их число конечно. В общем случае выполняется частичный перебор. Сокращение числа просмотренных вариантов достигается как за счет организации ветвления, так и за счет отсека подмножеств вариантов, не содержащих оптимального. Ветвление, т.е. разбиение множества вариантов на подмножества, выполняется по методу в ширину или в глубину с возвращением с заданным порядком построения ветвей. Реализация метода, прежде всего принцип разбиения множества вариантов на подмножества и вычисление условий отсека, существенно зависит от вида задачи, а степень сокращения перебора - от ее конкретных данных. В худшем случае метод может превратиться в полный перебор.

Организация ветвления подразумевает выбор в процессе разбиения наиболее перспективной вершины для ветвления, т.е. в разбиении того подмножества вариантов, к которому вероятнее всего принадлежит оптимальный. Таким образом, основные операции метода ветвей и границ - это ветвление и отсечение. Как было показано выше, в качестве оценки перспективности ветви используется верхняя или нижняя границы целевой функции, вычисляемые для каждого подмножества вариантов. В ряде случаев они могут использоваться и для отсека ветвей.



Можно выделить три основных способа отсечения ветвей.

1. Сравнение оценки (нижней – $F_H(M_i)$ или верхней – $F_B(M_i)$ границы) со значением целевой функции для уже найденного (опорного) решения $F_{оп}$. Действительно, если при решении задачи на минимум целевой функции $F_H(M_i) > F_{оп}$, то подмножество M_i не содержит оптимального варианта, и соответствующая вершина дерева решений отсекается. Опорное решение можно получить приближенным алгоритмом заранее либо в ходе решения задачи методом ветвей и границ. При разбиении пространства решений по методу в глубину с возвращением опорное решение получается на более ранних этапах построения дерева решений, чем при методе в ширину.

2. Сравнение двух оценок. Такое отсечение выполняется, если, например в задаче на минимум целевой функции, для подмножества вариантов соответствующей вершины можно найти оценку снизу $F_H(M_i)$ и сверху $F_B(M_i)$. Тогда, если для некоторого подмножества M_i окажется, что $F_H(M_i) > F_B(M_i)$, то ветвление в вершине, соответствующей M_i прекращается.

3. Прекращение ветвления, если соответствующее подмножество не содержит оптимального решения, можно установить, во-первых, в «особых точках» по значениям оценочной функции, например нижней границы (см. § 3.2), и, во-вторых, если известен оптимальный среди вариантов этого подмножества. Чем точнее будет получена оценка, т.е. чем ближе она будет к значению целевой функции для оптимального варианта подмножества, тем меньшее количество вершин дерева решений будет построено и исследовано.

Точность оценки зависит от принципа разбиения множества вариантов на подмножества и вида оценочной функции или способа ее вычисления. Следовательно, если возможно использование нескольких принципов разбиения, необходимо выбирать тот, при котором оценки более точны. Как правило, число отсечений тем больше, чем сильнее отличаются оценки подмножеств.

Таким образом, лучшими являются тот принцип разбиения и такая оценочная функция, при которых разность между оценками подмножеств наибольшая, а сами оценки вычисляются с наибольшей точностью в смысле их близости к значению целевой функции для оптимального варианта подмножества. Число отсечений зависит и от способа ветвления, хотя никаких точных оценок и рекомендаций указать нельзя.

Рассмотрим основные способы ветвления.

1. Разбиение множества вариантов на подмножества по методу «в ширину» и выбор вершины ветвления по минимуму (максимуму) оценочной функции. Сначала выполняется разбиение всего множества вариантов, т.е. строятся на следующем от корня уровне все или часть вершин его потомков. Затем на каждом шаге выбирается вершина ветвления по минимуму нижней границы (максимуму верхней для задачи на максимум целевой функции) и разбивается соответствующее ей подмножество вариантов. В ходе ветвления используется, если это возможно, тот или иной способ отсечения ветвей и вершин. Процесс выбора вершин и ветвления повторяется для всех висячих вершин, т.е. еще не отсеченных и не ставших концевыми в смысле их соответствия одному из вариантов решения задачи.

2. Разбиение множества вариантов по методу «в глубину с возвращением», т.е. последовательное построение ветвей. Строят полностью одну ветвь дерева решений, т.е. находят один вариант решения задачи. Полученное для этого варианта значение целевой функции может использоваться как отсекающая оценка. Далее ветвление выполняется последовательно от построенной ветви, начиная с вершины M_i , предшествовавшей конечной. При этом, если не происходит отсечения, новая ветвь достраивается до конца. Процедура повторяется по отношению к вершинам новой ветви. После построения всех возможных ветвей, проходящих через вершину, сопоставленную множеству M_i , отыскивается следующая вершина и процесс повторяется. При последовательном способе ветвления построение дерева решений может выполняться, начиная с левой ветви - слева направо, или с правой - справа налево. Количество построенных вершин дерева решений зависит от очередности развития ветвей.

3. Комбинация двух рассмотренных способов. Простейшая комбинация заключается в следующем. Строится одна ветвь. Далее развитие дерева решений происходит по методу «в ширину», полученное опорное решение используется для отсечения, выбор вершины ветвления выполняется по минимуму нижней границы в задачах на минимум целевой функции (по максимуму верхней).



В ходе решения, если возможно, уточняется значение отсекающей оценки. Оптимальное решение будет найдено, когда в задаче на минимум целевой функции значение для некоторой конечной вершины $F(M_i^k)$ меньше нижней границы $F_H(M_i)$ для всех висячих вершин и меньше значения целевой функции $F(\{M_i^k\})$ для всех остальных конечных вершин (в задаче на поиск максимума целевой функции соответственно «больше»).

Вычисление верхних и нижних границ и различные способы ветвления рассмотрим на примере нахождения простой цепи (маршрута) максимальной и минимальной длины, соединяющей некоторую исходную и конечную вершины в ориентированном графе G [10], показанном на *рис. 3.6, а*. Принцип разбиения множества вершин на подмножества тот же, что и выше, т.е. множество маршрутов, порождаемых множеством M_i разбиваем на подмножества маршрутов $\{M_j, \dots, M_k\}$, содержащих ребра графа $G \{(x_i, x_j), \dots, (x_j, x_k)\}$. На последующих рисунках в центре кружка, изображающего вершину дерева решений, будем указывать номер вершины графа G . Таким образом, ребро, соединяющее две вершины дерева решений, будет соответствовать ребру графа G , инцидентному таким же вершинам, а само дерево решений - это полное дерево простых цепей, соединяющих некоторые исходную и конечную вершины графа G . Около ребер проставим их длины (веса), сверху вершин дерева решений (слева или справа) - верхнюю и/или нижнюю границы, справа от вершины - номер вершины в порядке появления при построении дерева решений.



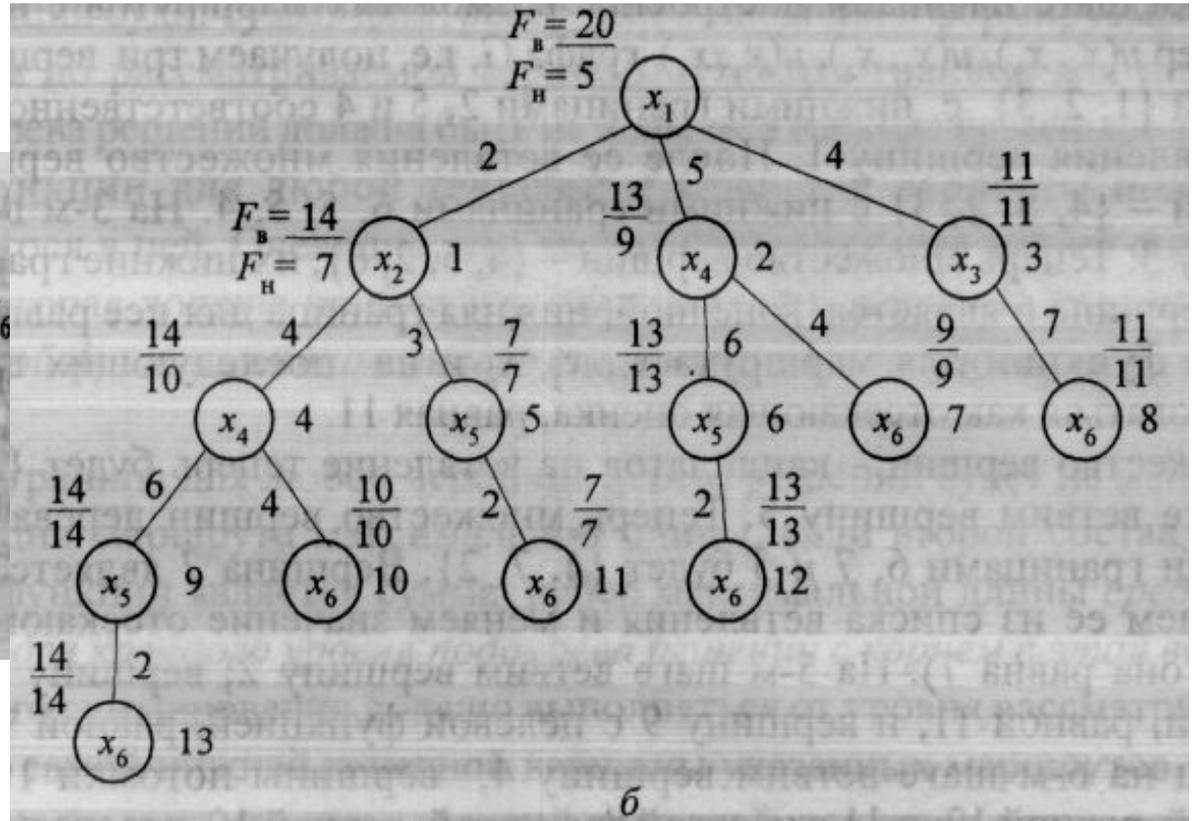
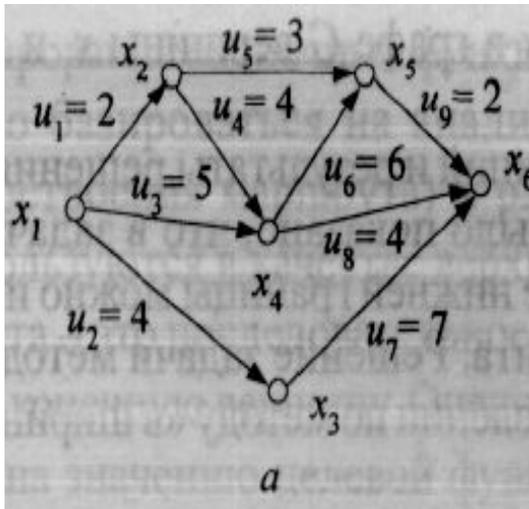


Рис. 3.6. Ориентированный граф (а) и дерево всех его простых цепей, построенное по методу в ширину (б)

Пусть исходной будет вершина x_1 , а конечной - x_6 . В данном графе простых цепей шесть:

$$\begin{aligned} c_1 &= \{x_1, x_2, x_5, x_6\}; \\ c_2 &= \{x_1, x_2, x_4, x_5, x_6\}; \\ c_3 &= \{x_1, x_2, x_4, x_6\}; \\ c_4 &= \{x_1, x_4, x_5, x_6\}; \\ c_5 &= \{x_1, x_4, x_6\}; \\ c_6 &= \{x_1, x_3, x_6\}. \end{aligned}$$

Дерево простых цепей, соединяющих в графе G вершины x_1 и x_6 показано на *рис. 3.6, б*.

Рассмотрим примеры оценочных функций и результаты решения задачи при различных способах ветвления. Выше было показано, что в задачах на поиск маршрута минимальной длины в качестве нижней границы можно использовать сумму длин ребер построенного фрагмента. Решение задачи методом ветвей и границ для графа G с $x_n = x_1$ и $x_k = x_6$ при ветвлении по методу «в ширину» показано в виде дерева решений на *рис. 3.7*.

Рассмотрим по шагам этот процесс.

На 1-м шаге начинаем построение возможных маршрутов с включения в них ребер $u(x_1, x_2)$, $u(x_1, x_4)$, $u(x_1, x_3)$ графа G , т.е. получаем три вершины дерева решений **{1, 2, 3}** с нижними границами **2, 5** и **4** соответственно. Выбираем для ветвления вершину **1**. После ее ветвления множество вершин дерева решений - **{4, 5, 2, 3}** с нижними границами **6, 5, 5, 4**. На 3-м шаге ветвим вершину **3**. Теперь множество вершин - **{4, 5, 2, 6}**, их нижние границы - **6, 5, 5, 11**. Вершина **6** является конечной, нижняя граница для нее равна значению целевой функции для маршрута x_1, x_3, x_6 и на последующих шагах будет использоваться как отсекающая оценка, равная **11**.



Множество вершин - кандидатов на ветвление теперь будет $\{4, 5, 2\}$. На 4-м шаге ветвим вершину 5. Теперь множество вершин дерева решений с нижними границами 6, 7 и 5 будет $\{4, 7, 2\}$. Вершина 7 является конечной, исключаем ее из списка ветвления и меняем значение отсекающей оценки (теперь она равна 7). На 5-м шаге ветвим вершину 2; вершину 8 с нижней границей, равной 11, и вершину 9 с целевой функцией, равной 9, отсекаем. Наконец на 6-м шаге ветвим вершину 4, вершины-потомки 10 с нижней границей, равной 12, и 11 с целевой функцией, равной 10, отсекаем. Решением является маршрут, соответствующей конечной вершине 7 – x_1, x_2, x_3, x_6 с суммой длин ребер, равной 7.

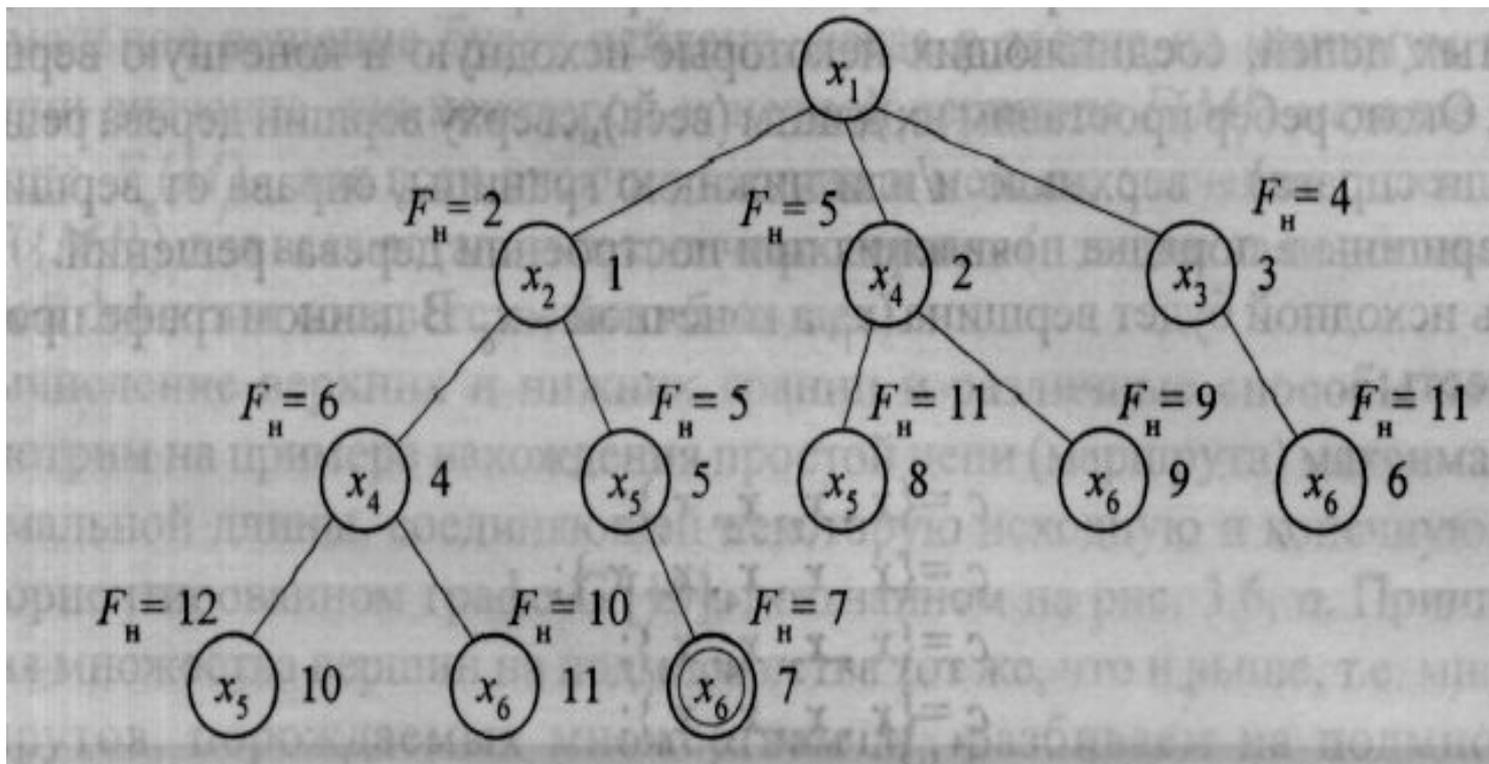


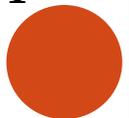
Рис. 3.7. Дерево поиска маршрута минимальной длины в графе G (см. рис. 3.6, а) методом ветвей и границ при ветвлении по нижней границе

Оценочная функция в виде суммы длин ребер, уже включенных в формируемый маршрут, является минимальным значением нижней границы.

Конструирование оценочных функций является сложной творческой задачей. Оно должно базироваться на анализе сущности задачи (в данном случае процесса построения маршрута) и рассмотренных в § 3.2 свойств функций, которые используются в качестве верхних и нижних границ. Процесс построения маршрута - это последовательное включение в него ребер до тех пор, пока не придем в конечную вершину. Оценочная функция в конечной вершине должна быть равна значению целевой функции, т.е. сумме длин ребер, составляющих соответствующий маршрут. Следовательно, одной из составляющих оценочной функции будет сумма длин ребер, уже вошедших в формируемый маршрут от корня до рассматриваемой вершины. Нижняя граница для некоторой вершины дерева решений должна быть не больше, а верхняя не меньше значения целевой функции для любой текущей и конечной вершины поддерева, начинающегося в ней. Очевидно, что вторая составляющая нашей оценочной функции, которая должна приблизить нижнюю или верхнюю границу к *значению целевой функции оптимального варианта*, должна удовлетворять этому условию.

Верхняя граница для любой вершины дерева решений будет не меньше целевой функции маршрута максимальной длины, если второй составляющей оценочной функции является сумма ребер максимальной длины *среди ребер маршрутов для каждого уровня поддерева решений с корнем в этой вершине*. Очевидно, что суммирование должно выполняться от уровня рассматриваемой вершины до самой нижней конечной вершины указанных маршрутов. Данная оценка действительно является верхней границей, так как в основу положено предположение, что на каждом шаге в строящийся маршрут включается ребро максимальной длины, тогда как в действительности это может быть ребро другого маршрута, и суммирование выполняется до максимально возможного конечного количества ребер маршрутов, проходящих через данную вершину. Например, для вершины 1 дерева решений, показанного на *рис. 3.6, б*,

$$F_B = 2 + \max\{4, 3\} + \max\{6, 4, 2\} + \max\{2\} = 14.$$



Сконструированная для оценки верхней границы функция не возрастает и ее значение в конечных вершинах равно значению целевой функции для соответствующего варианта маршрута.

Компонентами второй составляющей для нижней границы должны быть ребра минимальной длины для каждого уровня дерева решений среди ребер маршрутов, проходящих через эту вершину. Суммируя длины этих ребер от уровня рассматриваемой вершины до самой верхней, т.е. ближайшей к корню, конечной вершины этих маршрутов, получим вторую составляющую. Очевидно, что оценка не убывает и в конечной вершине равна значению целевой функции.

Сказанное выше не означает, что для получения таких оценок необходимо построить полностью дерево решений и сформировать все варианты маршрутов. Однако вычисление оценок этого типа по сравнению с оценкой в виде суммы длин ребер, включенных в фрагмент маршрута, требует выполнения довольно сложных преобразований. Рассмотрим пример вычисления нижней границы для вершины 2 (см. *рис. 3.6, б*). Пусть граф G задан аналитически через отображение множества X в себя, т.е. в форме $G(X, F)$. Здесь $X = \{x_i \mid i = \dots\}$ - множество вершин графа, $F = \{Fx_i \mid i = \dots\}$ - прямое отображение множества X в себя, т.е. $Fx_i = X$. $\subset X$ - множество тех вершин $x_j \in X$, в которые заходят дуги из вершины x_i . Каждому отображению Fx_i поставим во взаимно-однозначное соответствие вектор длин соответствующих дуг $L_i = \{l(x_i, x_j) \mid \forall x_j \in Fx_i\}$. Тогда первая компонента нижней границы вершины 2 определяется как $l(x_1, x_4) = 5$. Для определения второй составляющей находим $Fx_4 = \{x_5, x_6\}$ и выбираем минимум из $l(x_4, x_5)$ и $l(x_4, x_6)$. Таким образом, получим нижнюю границу для вершины 2, равную 9.

На *рис. 3.6, б* над вершинами (слева или справа) показана пара чисел в виде a/b , где a - верхняя граница; b - нижняя граница, рассчитанные по сконструированным выше оценочным функциям.



На рис. 3.8 представлено дерево решения задачи поиска маршрута максимальной длины по методу ветвей и границ при ветвлении в ширину (выбор вершины ветвления по максимуму верхней границы), а на рис. 3.9, а и б- при последовательном ветвлении, причем на рис. 3.9, а порядок построения ветвей справа налево, а на рис. 3.9, б - слева направо.

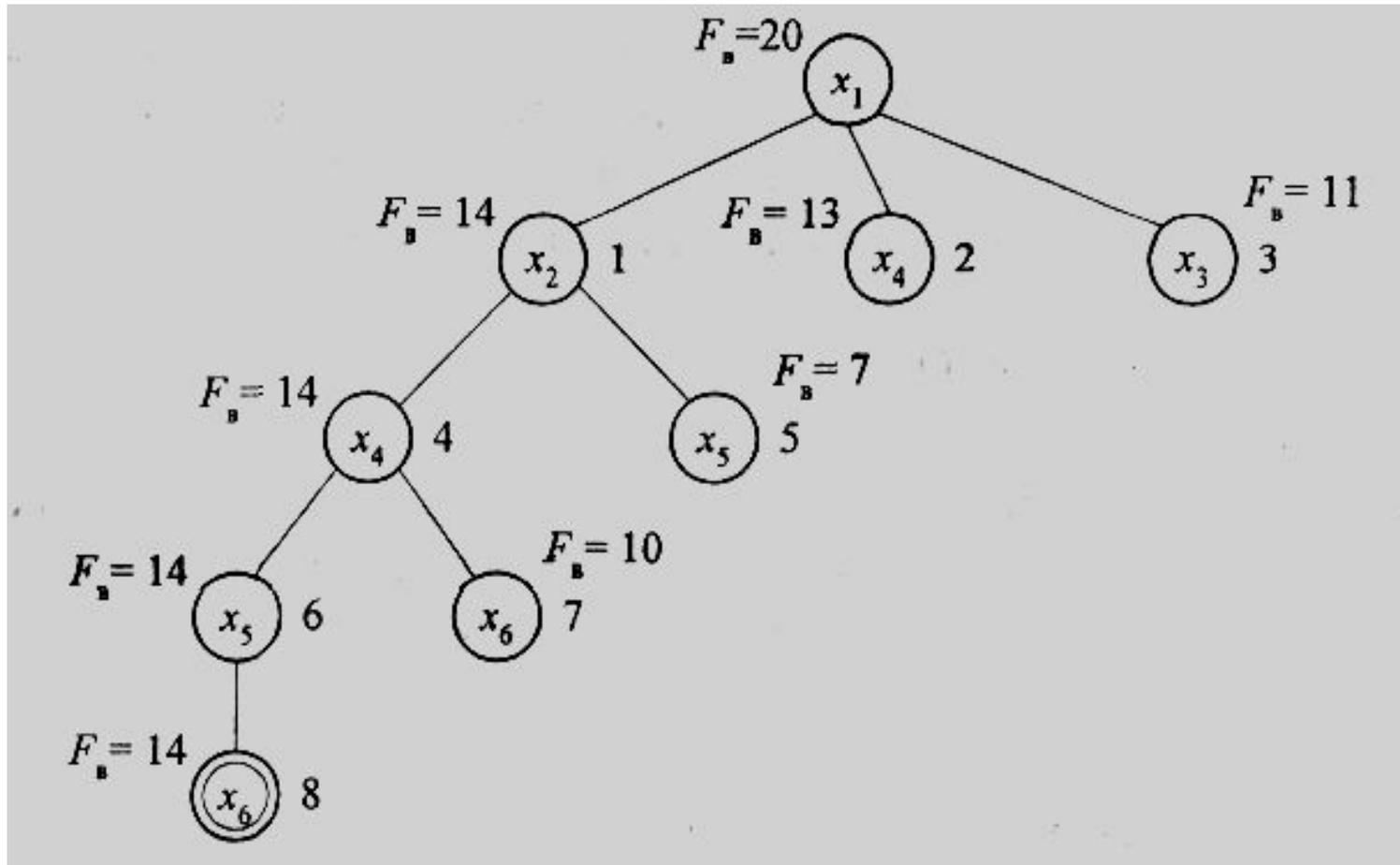


Рис. 3.8. Дерево поиска маршрута максимальной длины в графе G (см. рис. 3.6, а) по методу ветвей и границ, при ветвлении в ширину



Рис. 3.9. Дерево поиска маршрута максимальной длины в графе G (см. рис. 3.6, а) по методу ветвей и границ при последовательном построении ветвей справа налево (а)

На рис. 3.10 показано дерево решения при поиске маршрута минимальной длины по методу ветвей и границ при ветвлении в ширину. Выбор вершины ветвления выполняется по минимуму нижней границы сконструированной выше двухкомпонентной функции (сравните с рис. 3.7). На рис. 3.11, а и б представлены деревья решения той же задачи при последовательном построении ветвей (на рис. 3.11, а - справа налево и на рис. 3.11, б - слева направо).

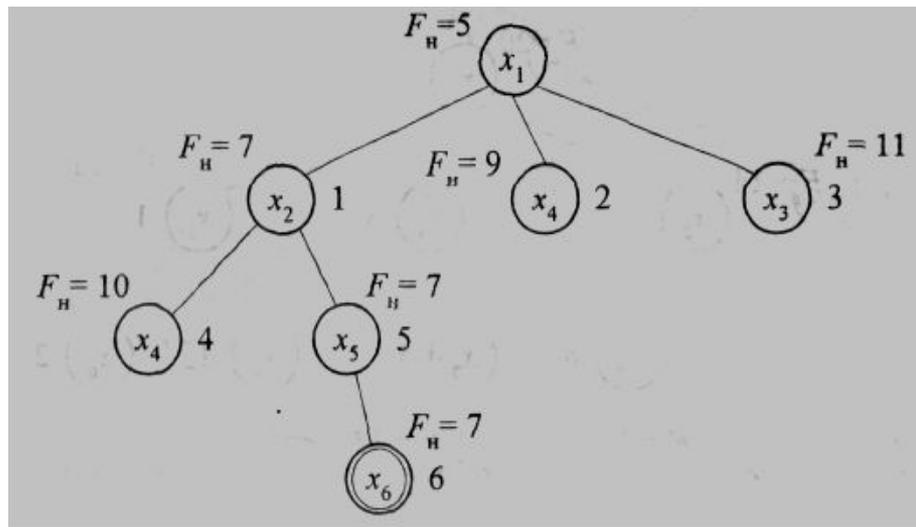


Рис. 3.10. Дерево поиска маршрута минимальной длины в графе G (см. рис. 3.6, а) по методу ветвей и границ при ветвлении по нижней границе

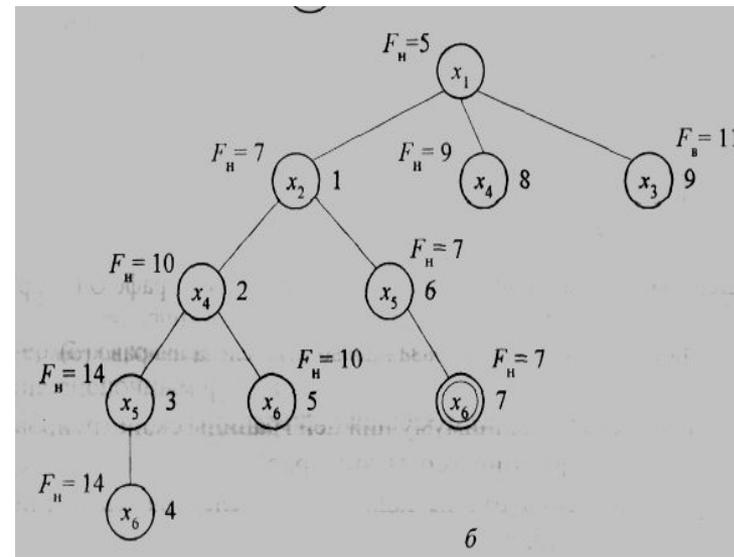
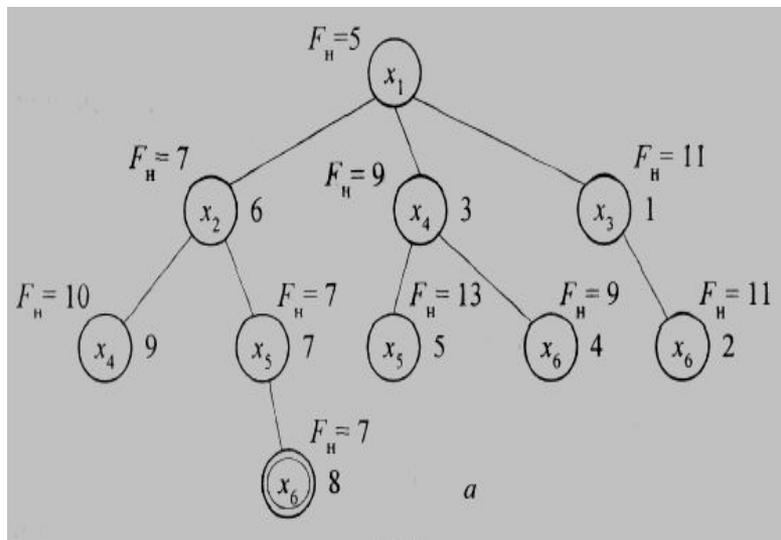


Рис. 3.11. Деревья поиска маршрута минимальной длины в графе G (см. рис. 3.6, а) по методу ветвей и границ при последовательном построении ветвей справа налево (а) и слева направо (б)