Лабораторная работа № 4

Метод ветвей и границ. Задача коммивояжера

Цель работы: Изучить метод ветвей и границ для решения задачи о коммивояжёре дискретного программирования и применить его на практическом примере.

Теоретические сведения

Пусть $M = \{m_1,...,m_r\}$ - конечное множество и $f : M \to \Re$ - вещественнозначная функция на нем. Требуется найти минимум этой функции и элемент множества, на котором этот минимум достигается.

Когда имеется та или иная дополнительная информация о множестве, решить эту задачу иногда удается без полного перебора элементов всего множества *M*. Однако чаще всего требуется полный перебор. В этом случае обязательно возникает вопрос, как лучше организовать перебор.

Метод ветвей и границ применим в том случае, когда выполняются специфические дополнительные условия на множество *M* и минимизируемую на нем функцию. Так, *предположим*, что имеется вещественно-значная функция φ на подмножестве множества *M* со следующими двумя свойствами:

1) для $\forall i \ \varphi(\{m_i\}) = f(m_i)$,

где $\{m_i\}$ - множество, состоящее из единственного элемента m_i;

2) если $U,V \subseteq M$ и $U \subseteq V$, то $\varphi(U) \geq \varphi(V)$.

В этих условиях можно организовать перебор элементов множества *M* с целью минимизации функции на нем так: разобьем множество *M* на части (любым способом) и выберем ту из его частей Ω_1, на которой функция φ минимальна; затем разобьем на несколько частей множество Ω_1 и выберем ту из его частей Ω_2, на которой функция φ минимальна; потом разобьем Ω_2 на несколько частей и выберем ту из них, где φ минимальна, и так далее, пока не придем к какому-либо одноэлементному множеству $\{m_i\}$. Это одноэлементное множество $\{m_i\}$ называется *рекордом*. Функция φ, которой мы пользуемся при выборе, называется *оценочной.*

Очевидно, что рекорд не обязан доставлять минимум функции *f*, однако при благоприятных обстоятельствах возникает возможность сократить перебор.

Описанный выше процесс построения рекорда состоял из последовательных этапов, на каждом из которых фиксировалось несколько множеств и затем выбиралось одно из них.

Пусть $A_1,\dots, A_s$ - подмножества множества M, возникшие на предпоследнем этапе построения рекорда, и пусть множество A_1 оказалось выбранным с помощью оценочной функции. Именно при разбиении A_1 и возник рекорд, который для определенности обозначим через $\{m_i\}$. Согласно сказанному выше, $\varphi(A_1) \le \varphi(A_i)$, $i = 1,\dots, s$; кроме того, по определению оценочной функции, $\varphi(A_1) \le \varphi(\{m_1\}) = f(m_1)$.

Предположим, что $f(m_1) \le \varphi(A_2)$. Тогда для любого элемента m множества M, принадлежащего множеству A_2, будут верны неравенства $f(m_1) \le \varphi(A_2) \le f(m)$. Это значит, что при полном переборе элементов из M элементы из A_2 уже вообще не надо рассматривать. Если же неравенство $f(m_1) \le \varphi(A_2)$ не будет выполнено, то все элементы из A_2 надо последовательно сравнить с найденным рекордом и, как только отыщется элемент, дающий меньшее значение оптимизируемой функции, заменить им рекорд и продолжить перебор. Последнее действие называется *улучшением рекорда*.

Метод ветвей и границ состоит в естественной графической интерпретации всего изложенного: строится многоуровневое дерево, на нижнем уровне которого располагаются элементы множества M. Ветви на дереве ведут к рекорду и его улучшениям, при этом часть ветвей остается «отсеченной», потому что их развитие оказалось нецелесообразным.

Пример. Решить задачу коммивояжера.

Пусть имеется несколько городов, некоторым образом соединенных дорогами с известной длиной. Требуется установить, имеется ли кратчайший путь, двигаясь по которому, можно побывать в каждом городе только один раз и вернуться в город, откуда путь был начат («обход коммивояжера»).

Формализуем условие задачи в терминах теории графов. Города будут вершинами графа, а дороги между городами - ориентированными (направленными) ребрами графа, на каждом из которых задана весовая функция: вес ребра - это длина соответствующей дороги; путь, который требуется найти, - это ориентированный *остовный* простой цикл минимального веса в орграфе (напомним: цикл называется *остовным*, если он проходит по всем вершинам графа; цикл называется *простым*, если он проходит по каждой своей вершине только один раз; цикл называется *ориентированным*, если начало каждого последующего ребра совпадает с концом предыдущего ребра; *вес* цикла - это сумма весов его ребер; наконец, орграф называется *полным*, если в нем имеются все возможные ребра); такие циклы называются также *гамильтоновыми.*

Очевидно, что в полном орграфе гамильтоновые циклы есть. Заметим, что вопрос о наличии в орграфе гамильтонова цикла достаточно рассмотреть как частный случай задачи коммивояжера для полных орграфов. Действительно, если данный орграф не является полным, то его можно дополнить до полного недостающими ребрами и каждому из добавленных ребер приписать вес ∞, считая, что ∞ - это «компьютерная бесконечность», т.е. максимальное из всех возможных чисел. Если во вновь построенном полном орграфе найти теперь наименьший гамильтонов цикл, то при наличии у него ребер с весом ∞ можно будет говорить, что в данном исходном графе «цикла коммивояжера» нет. Если же в полном орграфе легчайший гамильтонов цикл окажется конечным по весу, то он и будет искомым циклом в исходном графе. Отсюда следует, что задачу коммивояжера достаточно решить для полных орграфов с весовой функцией.

Сформулируем теперь задачу коммивояжера в окончательном виде: пусть $G = [A,B]$ - полный ориентированный граф и $v: B \to \Re$ - весовая функция. Требуется найти простой остовный ориентированный цикл («цикл коммивояжера») минимального веса.

Пусть $A = \{a_1, \dots, a_p\}$ - конкретный состав множества вершин и $M = (m_{ij})$, $i, j = 1, \dots, p$ - весовая матрица данного орграфа, т.е. $m_{ij} = v(a_i, a_j)$, причем для любого i $m_{ij} = \infty$.

Алгоритм решения задачи коммивояжера

Первый шаг. Зафиксируем множество всех обходов коммивояжера (т.е. всех простых ориентированных остовных циклов). Поскольку граф - полный, это множество заведомо непустое. Сопоставим ему число, которое на этом множестве будет играть роль значения оценочной функции. Это число равно сумме констант приведения данной матрицы весов ребер графа. Если множество всех обходов коммивояжера обозначим через Γ, то сумму констант приведения матрицы весов обозначим через φ(Γ). Приведенную матрицу весов данного графа следует запомнить; обозначим ее через M_1. Таким образом, итог первого шага: множеству Γ всех обходов коммивояжера сопоставлены число φ(Γ) и матрица M_1.

Второй шаг. Выберем в матрице M_1 самый тяжелый нуль (функцию штрафа); пусть он стоит в клетке (i, j). Зафиксируем ребро графа (i, j) и разделим множество Γ на две части: $\Gamma_{\{(i,j)\}}$, состоящую из обходов, которые проходят через ребро (i, j), и $\Gamma_{\overline{\{(i,j)\}}}$, состоящую из обходов, которые не проходят через ребро (i, j). Сопоставим множеству $\Gamma_{\{(i,j)\}}$ матрицу $M_{1,1}$. Для этого в матрице M_1 заменим на ∞ число в клетке (i, j). Затем в полученной матрице вычеркнем строку под номером i и столбец под номером j, причем у оставшихся строк и столбцов сохраним их исходные номера.
Наконец, приведем эту последнюю матрицу и запомним сумму констант приведения.

Полученная приведенная матрица и будет матрицей $M_{1,1}$. Запомненную сумму констант приведения прибавим к φ(Γ) и результат, обозначаемый в дальнейшем через $\varphi(\Gamma_{\{(i,j)\}})$, сопоставим множеству $\Gamma_{\{(i,j)\}}$.

Теперь множеству $\Gamma_{\{\overline{(i,j)}\}}$ сопоставим некую матрицу $M_{1,2}$. Для этого в матрице M_1 заменим на ∞ число в клетке (i, j) и полученную в результате матрицу приведем. Сумму констант приведения запомним, а полученную матрицу обозначим через $M_{1,2}$. Прибавим запомненную сумму констант приведения к числу φ(Γ) и результат, обозначаемый в дальнейшем через $\varphi(\Gamma_{\{\overline{(i,j)}\}})$, сопоставим множеству $\Gamma_{\{\overline{(i,j)}\}}$. Выберем из множеств $\Gamma_{\{(i,j)\}}$ и $\Gamma_{\{\overline{(i,j)}\}}$ то, на котором минимальна функция φ (т.е. которому соответствует меньшее из чисел $\varphi(\Gamma_{\{(i,j)\}})$ и $\varphi\,\Gamma_{\{\overline{(i,j)}\}}$.

Заметим, что в проведенных рассуждениях использовался в качестве исходного только один фактический объект - приведенная матрица весов данного орграфа. По ней было выделено определенное ребро графа и построены новые матрицы, к которым, конечно, можно применить все то же самое.

При каждом таком повторном применении будет фиксироваться очередное ребро графа. Условимся о следующем действии: перед тем, как в очередной матрице вычеркнуть строку и столбец, в ней надо заменить на ∞ числа во всех тех клетках, которые соответствуют ребрам, заведомо не принадлежащим тем гамильтоновым циклам, которые проходят через уже отобранные ранее ребра.

К выбранному множеству с сопоставленными ему матрицей и числом φ применим, до тех пор пока это возможно, указанный выше алгоритм улучшения рекорда вплоть до получения окончательного ответа.

Можно доказать, что в результате получится множество, состоящее из единственного «обхода коммивояжера», вес которого равен очередному значению функции φ. Таким образом, оказываются выполненными все условия, обсуждавшиеся при описании метода ветвей и границ.

Пример решения задачи коммивояжера

Решить методом ветвей и границ задачу коммивояжера для графа, содержащего 5 вершин.

Пусть исходный ориентированный граф задан матрицей стоимости:

$$C_1 = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \infty & 68 & 73 & 24 & 70 & 9 \\ 2 & 58 & \infty & 16 & 44 & 11 & 92 \\ 3 & 63 & 9 & \infty & 86 & 13 & 18 \\ 4 & 17 & 34 & 76 & \infty & 52 & 70 \\ 5 & 60 & 18 & 3 & 45 & \infty & 58 \\ 6 & 16 & 82 & 11 & 60 & 48 & \infty \end{array}$$

Начальное приведение матрицы стоимости. Пусть имеется некоторая числовая матрица. *Привести строку* этой матрицы означает выделить в строке минимальный элемент (его называют *константой приведения*) и вычесть его из всех элементов этой строки. В результате в этой строке на месте минимального элемента окажется нуль, а все остальные элементы будут неотрицательными. Матрица стоимости называется *приведенной*, если она имеет в каждой строке и каждом столбце хотя бы один нуль. Сумма констант приведения образует нижнюю граничную оценку стоимости любого возможного тура.

Вычисление функции штрафа. Функция штрафа - это множество чисел, вычисленных для каждого нуля приведенной матрицы посредством суммирования двух минимальных чисел из той строки и из того столбца, в которых расположен нулевой элемент.

Выбор ребра ветвления. Для ветвления необходимо выбирать ребро, которому соответствует максимальная функция штрафа. Если существует несколько одинаковых максимальных значений функции штрафа, выбор среди них может быть произвольным.

Весом элемента матрицы называют сумму констант приведения матрицы, которая получается из данной матрицы заменой анализируемого элемента на ∞. Следовательно, выражение *самый тяжелый нуль* в матрице означает, что в матрице подсчитан вес каждого нуля, а затем фиксирован нуль с максимальным весом.

Вычисление граничной оценки для ветви, соответствующей невключению ребра в тур. Эта оценка вычисляется как сумма граничной оценки, соответствующей предыдущему узлу дерева перебора, и выбранного значения функции штрафа.

Вычисление граничной оценки для ветви, соответствующей включению ребра в тур. Для вычисления этой граничной оценки необходимо:

- вычеркнуть в матрице стоимости строку и столбец, соответствующие выбранному ребру;
- скорректировать полученную матрицу таким образом, чтобы устранить возможность досрочного завершения тура (устранить циклы);
- сделать приведение (если необходимо) полученной матрицы, и если константа приведения отлична от нуля, сложить эту константу с граничной оценкой предыдущего узла.

Проверка на окончание решения. Если скорректированная матрица имеет размер 2×2, а узел дерева, которому соответствует эта матрица, имеет минимальную граничную оценку, то решение задачи заканчивается: два оставшихся нуля этой матрицы соответствуют двум последним ребрам, которые включаются в тур непосредственно, при этом стоимость тура не изменяется.

Решение

Сначала приводим исходную матрицу по строкам (матрица C_1):

$$C_1 = \begin{array}{c|cccccc|c} & 1 & 2 & 3 & 4 & 5 & 6 & h_i \\ \hline 1 & \infty & 59 & 64 & 15 & 61 & 0 & 9 \\ 2 & 47 & \infty & 5 & 33 & 0 & 81 & 11 \\ 3 & 54 & 0 & \infty & 77 & 4 & 9 & 9 \\ 4 & 0 & 17 & 59 & \infty & 35 & 53 & 17 \\ 5 & 57 & 15 & 0 & 42 & \infty & 55 & 3 \\ 6 & 5 & 71 & 0 & 49 & 37 & \infty & 11 \end{array}$$

В последнем столбце этой матрицы h_i записаны константы приведения по строкам.

Полученную матрицу необходимо привести по столбцам. В результате получаем матрицу $C_1^{'}$, в которой в столбце h_j записаны константы приведения по столбцам:

$$C_1^{'} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \infty & 59 & 64 & 0 & 61 & 0 \\ 2 & 47 & \infty & 5 & 18 & 0 & 81 \\ 3 & 54 & 0 & \infty & 62 & 4 & 9 \\ 4 & 0 & 17 & 59 & \infty & 35 & 53 \\ 5 & 57 & 15 & 0 & 27 & \infty & 55 \\ 6 & 5 & 71 & 0 & 34 & 37 & \infty \\ \hline h_j & 0 & 0 & 0 & 15 & 0 & 0 \end{array}$$

Сумма констант приведения H дает нижнюю граничную оценку стоимости всех туров:

$$H = \sum_i h_i + \sum_j h_j = 9 + 11 + 9 + 17 + 3 + 11 + 15 = 75.$$

Вычисляем для ребер, помеченных нулем в матрице $C_1^{'}$, значения функций штрафа, которые обозначаем символом D_{ij}:

$D_{16} = 0 + 9 = 9$; $D_{41} = 17 + 5 = 22$;

$D_{25} = 5 + 4 = 9$; $D_{53} = 15 + 0 = 15$;

$D_{32} = 4 + 15 = 19$; $D_{63} = 5 + 0 = 5$.

Максимальное значение функции штрафа равно 22, на основании чего в качестве ребра ветвления выбираем ребро (4,1). Разбиваем множество всех туров R на два подмножества: {4,1} - подмножество всех туров, в которое входит ребро (4,1), и подмножество всех туров (4,1*), в которое ребро (4,1) не входит. Строим первый фрагмент ограниченного дерева перебора (рис.5). При этом граничная оценка в ветви (4,1*) находится непосредственно как сумма
75 + 22 = 97.

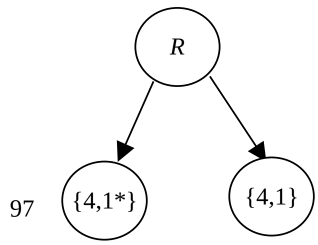

Рис.5. Первый фрагмент ограниченного дерева перебора

Вычеркиваем в матрице $C_1^{'}$ четвертую строку и первый столбец, а чтобы запретить досрочное завершение тура по ребру графа (1,4), весу этого ребра присваиваем значение ∞. В результате подобной корректировки получаем матрицу C_2:

$$C_2 = \begin{array}{c|ccccc} & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 59 & 64 & \infty & 61 & 0 \\ 2 & \infty & 5 & 18 & 0 & 81 \\ 3 & 0 & \infty & 62 & 4 & 9 \\ 5 & 15 & 0 & 27 & \infty & 55 \\ 6 & 71 & 0 & 34 & 37 & \infty \\ \hline h_j & 0 & 0 & 18 & 0 & 0 \end{array}$$

Ее необходимо привести по четвертому столбцу (константа приведения $h_4 = 18$), в результате чего получаем приведенную матрицу $C_2^{'}$. Вычисляем граничную оценку в ветви {4,1}, которая будет равна 75 + 18 = 93 (рис.6).

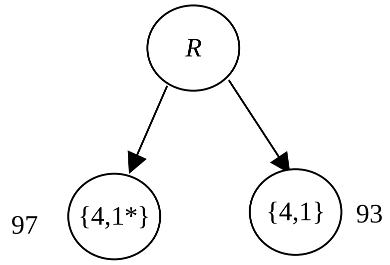

Рис.6. Первый фрагмент ограниченного дерева перебора с граничными оценками в листьях

Сравниваем оценки в листьях и приходим к выводу, что ветвление целесообразно продолжать в листе с оценкой 93. Так как этому листу соответствует матрица $C_2^{'}$, то с ней необходимо поступать как с исходной матрицей, т.е. вычислять для нее значения функций штрафа, выбирать максимальное значение штрафа и на его основе выбирать следующее ребро ветвления:

$$C_2^{'} = \begin{array}{c|ccccc} & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 59 & 64 & \infty & 61 & 0 \\ 2 & \infty & 5 & 0 & 0 & 81 \\ 3 & 0 & \infty & 44 & 4 & 9 \\ 5 & 15 & 0 & 9 & \infty & 55 \\ 6 & 71 & 0 & 16 & 37 & \infty \end{array}$$

Для матрицы $C_2^{'}$ имеем следующие значения функций штрафа:

$D_{16} = 59 + 9 = 68$; $D_{32} = 4 + 15 = 19$;

$D_{24} = 0 + 9 = 9$; $D_{53} = 9 + 0 = 9$;

$D_{25} = 0 + 4 = 4$; $D_{32} = 16 + 0 = 16$.

Выбираем максимальное значение 68, которому соответствует ребро ветвления (1,6), после чего можно построить второй фрагмент ограниченного дерева перебора (рис.7).

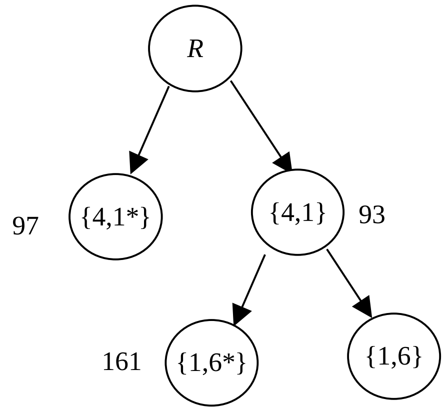

Рис.7. Второй фрагмент ограниченного дерева перебора

Теперь приступаем к вычислению граничной оценки в узле дерева (1,6). Для этого вычеркиваем в матрице $C_2^{'}$ первую строку и шестой столбец, а также предпринимаем меры для исключения досрочного завершения тура матрицы C_3.

В самом деле, к данному моменту в тур включено ребро (4,1) и есть предложение включить в него ребро (1,6). Из рис.8 видно, что досрочно завершить тур могли бы ребра (6,1) и (6,4).

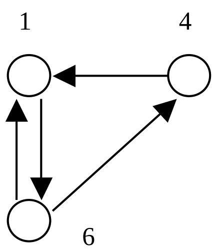

Рис.8. Пояснение к исключению досрочного завершения тура

Однако элемент (6,1) в матрице C_3 уже отсутствует, следовательно, эту «роль» может выполнить лишь элемент (6,4), поэтому заменяем его символом ∞. После чего окончательно матрица C_3 будет иметь вид:

$$C_3 = C_3^{'} = \begin{array}{c|cccc} & 2 & 3 & 4 & 5 \\ \hline 2 & \infty & 5 & 0 & 0 \\ 3 & 0 & \infty & 44 & 4 \\ 5 & 15 & 0 & 9 & \infty \\ 6 & 71 & 0 & \infty & 37 \end{array}$$

Матрица C_3 оказалась приведенной (поэтому ее можно, как это было ранее принято для приведенных матриц, пометить штрихом), а следовательно, оценка в узле (1,6) остается неизменной (рис.9).

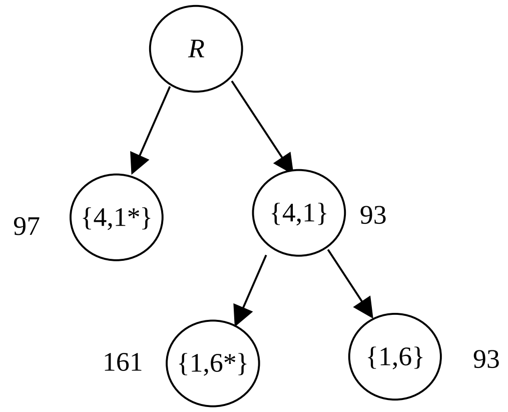

Рис.9. Второй фрагмент ограниченного дерева перебора
с граничными оценками в листьях

Вычисляем значения функций штрафа для матрицы C_3':

$D_{24} = 0 + 9 = 9$; $D_{53} = 9 + 0 = 9$;

$D_{25} = 0 + 4 = 4$; $D_{63} = 37 + 0 = 37$;

$D_{32} = 4 + 15 = 19$.

На основе вычисленных значений определяем следующее ребро ветвления (ребро (6,3)) и вычисляем граничную оценку в том узле дерева перебора, которому соответствуют туры, не содержащие ребра (6,3) (рис.10). При этом ветвление продолжается от узла с минимальной оценкой, равной 93.

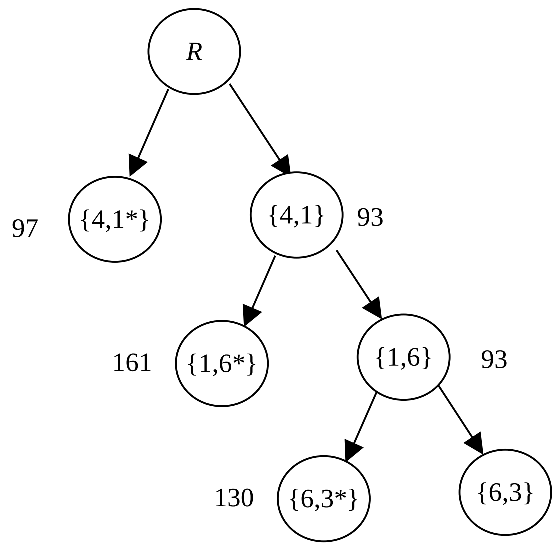

Рис.10. Третий фрагмент ограниченного дерева

Вычеркиваем в матрице $C_3^{'}$ шестую строку и третий столбец и корректируем ее на исключение досрочного завершения тура. Это достигается присваиванием символа бесконечности элементу (3,4). После чего получаем матрицу C_4

$$C_4 = \begin{array}{c|ccc} & 2 & 4 & 5 \\ \hline 2 & \infty & 0 & 0 \\ 3 & 0 & \infty & 4 \\ 5 & 15 & 9 & \infty \end{array}$$

и после ее приведения - матрицу $C_4^{'}$

$$C_4^{'} = \begin{array}{c|ccc} & 2 & 4 & 5 \\ \hline 2 & \infty & 0 & 0 \\ 3 & 0 & \infty & 4 \\ 5 & 6 & 0 & \infty \end{array}$$

Так как матрица C_4 не была приведена, то граничная оценка в узле (6,3) увеличивается по сравнению с предыдущим узлом ветвления на 9 и становится равной 102. Третий фрагмент ограниченного дерева перебора с граничными оценками в листьях будет иметь вид, изображенный на рис.11.

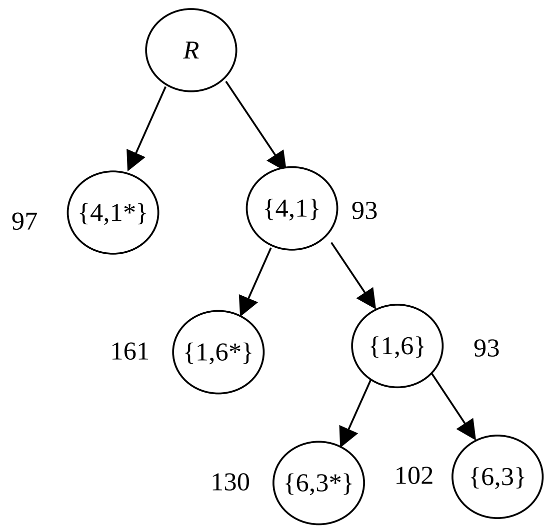

Рис.11. Третий фрагмент ограниченного дерева с граничными оценками в листьях

Рассмотрение дерева, приведенного на рис.11, показывает, что ветвление нужно продолжать из узла (4,1*), так как граничная оценка там меньше (97), чем в узле, из которого ветвление осуществлялось до сих пор. Подобная ситуация называется *перескоком*, и в этом случае может оказаться, что все полученные до сих пор данные будут утеряны.

Итак, переходим к узлу (4,1*), которому соответствует матрица C_5:

$$C_5 = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \infty & 68 & 73 & 24 & 70 & 9 \\ 2 & 58 & \infty & 16 & 44 & 11 & 92 \\ 3 & 63 & 9 & \infty & 86 & 13 & 18 \\ 4 & \infty & 34 & 76 & \infty & 52 & 70 \\ 5 & 60 & 18 & 3 & 45 & \infty & 58 \\ 6 & 16 & 82 & 11 & 60 & 48 & \infty \end{array}$$

Матрица C_5 не приведена, и после выполнения операции приведения получаем матрицу $C_5^{'}$:

$$C_5^{'} = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \infty & 59 & 64 & 0 & 61 & 0 \\ 2 & 42 & \infty & 5 & 18 & 0 & 81 \\ 3 & 49 & 0 & \infty & 62 & 4 & 9 \\ 4 & \infty & 0 & 42 & \infty & 18 & 36 \\ 5 & 52 & 15 & 0 & 27 & \infty & 55 \\ 6 & 0 & 71 & 0 & 34 & 37 & \infty \end{array}$$

У матрицы C_5 сумма констант приведения равна 97, что соответствует вычисленной ранее через функцию штрафа граничной оценке в узле (4,1*). Этот факт подтверждает правильность выполненных вычислений.

Вычисляем для матрицы $C_5^{'}$ значения функций штрафа:

$D_{14} = 0 + 18 = 18$; $D_{42} = 18 + 0 = 18$;

$D_{16} = 0 + 9 = 9$; $D_{33} = 15 + 0 = 15$;

$D_{25} = 5 + 4 = 9$; $D_{61} = 0 + 42 = 42$;

$D_{32} = 4 + 0 = 4$; $D_{63} = 0 + 0 = 0$.

Очередным ребром ветвления из узла (4,1*) будет ребро (6,1), так как ему соответствует максимальное значение функции штрафа. После корректировки матрицы $C_5^{'}$ (ребру (1,6) приписывается бесконечный вес) и вычисления граничных оценок окончательный четвертый фрагмент дерева перебора будет иметь вид, представленный на рис.12.

Из рис.12 видно, что минимальная граничная оценка (102) имеет место в узле (6,3), поэтому целесообразно осуществить обратный «перескок» в этот узел и продолжать дальнейшее ветвление из него. Этому узлу соответствует ранее вычисленная матрица $C_4^{'}$; дадим ей для удобства сквозной нумерации новое обозначение: $C_6 = C_6^{'}$.

Вычисляем для матрицы $C_6^{'}$ значения функций штрафа:

$D_{24} = 0 + 0 = 0$; $D_{32} = 4 + 6 = 10$;

$D_{25} = 0 + 4 = 4$; $D_{54} = 6 + 0 = 6$.

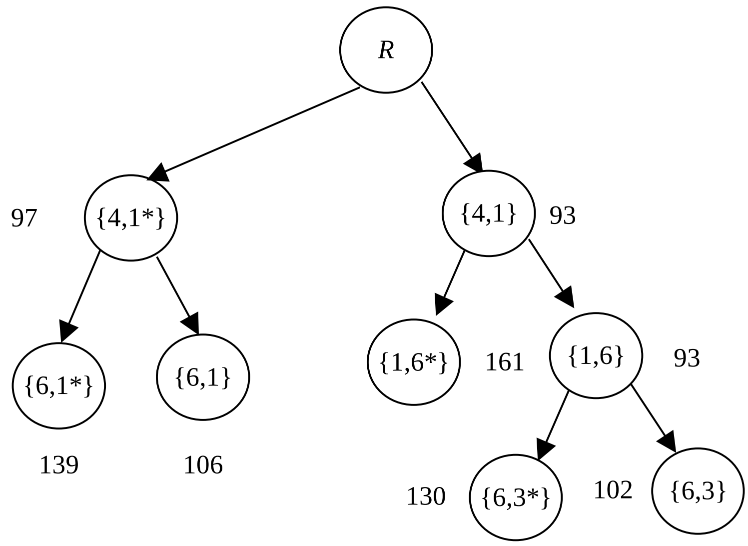

Рис.12. Четвертый фрагмент ограниченного дерева
с граничными оценками в листьях

Максимальное значение функции штрафа, равное 10, связано с ребром (3,2), поэтому выбираем его в качестве ребра ветвления, продолжая построение дерева из листа с граничной оценкой 102.

После исключения из матрицы $C_6^{'}$ третьей строки, второго столбца, а также ее корректировки получаем приведенную матрицу C_7:

$$C_7 = \begin{array}{c|cc} & 4 & 5 \\ \hline 2 & \infty & 0 \\ 5 & 0 & \infty \end{array}$$

Так как матрица C_7 уже приведена и имеет размер 2×2, то ребра (2,5) и (5,4) непосредственно включаем в тур, при этом граничная оценка 102 не изменяется.

Окончательный вид тура представлен на рис.13, а ограниченное дерево перебора - на рис.14. Легко проверить, что сумма весов ребер, вошедших в тур, равна 102, что подтверждает правильность полученного решения.

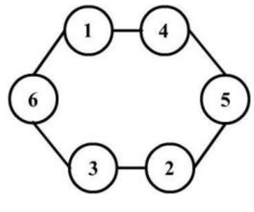

Рис.13. Оптимальный тур

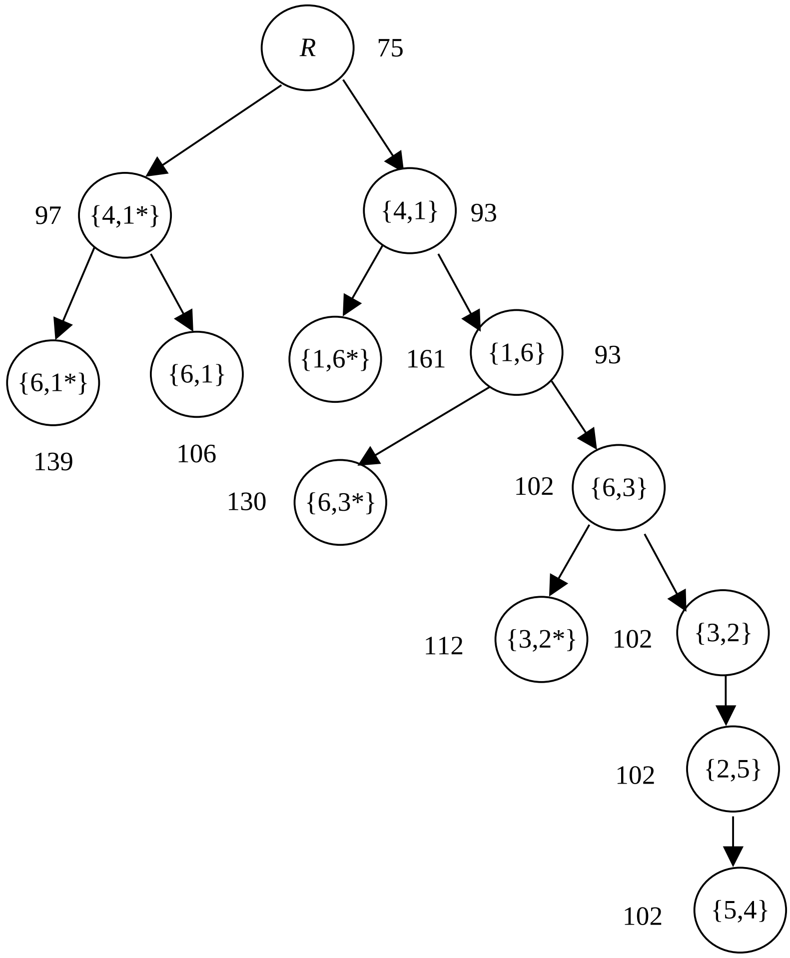

Рис.14. Окончательный вид ограниченного дерева перебора

Лабораторное задание

1. Решить задачу о коммивояжёре согласно варианту
2. Написать программу по алгоритму метода ветвей и границ.
3. Проверить верность решения задачи используя любой открытый ресурс. (онлайн или используя пакет прикладных программ)

Требования к отчету

Отчет должен содержать:

1) титульный лист, цель работы, краткие теоритические сведения, задачу;
2) результаты вычислений для каждого шага;
3) итоговое дерево решений;
4) результаты решения задачи коммивояжера;
5) скриншот проверки задачи (онлайн или используя пакет прикладных программ);
6) выводы по работе.

Варианты:

1)	∞	6	9	2	4
	6	∞	2	6	2
	6	3	∞	3	3
	5	7	3	∞	6
	3	2	2	5	∞

2)	∞	9	12	7	10
	9	∞	7	2	4
	2	4	∞	3	5
	8	7	1	∞	5
	3	4	3	5	∞

3)	∞	3	8	2	4
	2	∞	1	4	3
	2	2	∞	4	5
	3	6	4	∞	2
	9	1	10	5	∞

4)	∞	2	7	8	4
	3	∞	2	2	8
	9	8	∞	3	3
	6	1	1	∞	7
	3	5	2	4	∞

5)	∞	6	1	3	6
	1	∞	5	7	3
	2	7	∞	4	1
	3	1	9	∞	3
	1	4	5	3	∞

6)	∞	4	6	3	4
	5	∞	2	5	4
	9	2	∞	2	3
	3	3	2	∞	9
	1	7	3	2	∞

7)	∞	1	4	2	6
	3	∞	6	8	5
	3	9	∞	3	2
	1	1	4	∞	5
	8	4	2	4	∞

8)	∞	4	7	9	4
	3	∞	1	3	2
	10	9	∞	1	7
	4	6	1	∞	9
	5	2	3	7	∞

9)	∞	3	7	3	8
	9	∞	2	2	4
	7	9	∞	9	3
	4	7	3	∞	9
	5	1	5	5	∞

10)	∞	3	1	2	3
	8	∞	3	4	3
	3	1	∞	6	2
	1	8	9	∞	4
	3	7	4	1	∞

11)	∞	8	4	7	3
	9	∞	3	1	8
	4	9	∞	4	7
	3	3	6	∞	5
	6	4	2	6	∞

12)	∞	4	8	5	3
	6	∞	4	5	2
	1	5	∞	9	2
	11	3	12	∞	4
	2	3	7	6	∞

13)	∞	4	9	2	1
	9	∞	1	11	5
	7	4	∞	1	3
	7	6	3	∞	3
	2	11	4	4	∞

14)	∞	2	10	5	3
	8	∞	7	7	4
	4	3	∞	4	3
	1	6	3	∞	5
	5	4	8	4	∞

15)	∞	6	3	1	6
	4	∞	3	5	3
	9	3	∞	4	4
	2	6	2	∞	7
	3	1	1	9	∞

16)	∞	8	3	9	2
	4	∞	9	3	2
	6	1	∞	4	3
	2	3	8	∞	4
	4	1	1	9	∞

17)	∞	1	3	1	3
	2	∞	9	4	2
	9	3	∞	7	5
	7	2	1	∞	7
	1	4	6	3	∞

18)	∞	5	2	3	9
	3	∞	3	4	3
	1	8	∞	4	8
	2	3	2	∞	5
	4	5	7	2	∞

19)	∞	4	1	3	6
	7	∞	2	4	4
	4	9	∞	1	5
	8	1	4	∞	3
	3	9	2	3	∞

20)	∞	5	3	5	6
	1	∞	6	9	4
	5	2	∞	10	1
	3	6	4	∞	9
	4	5	2	3	∞

21)	∞	6	8	2	4
	1	∞	6	4	3
	9	3	∞	6	4
	5	5	2	∞	7
	3	2	1	3	∞

22)	∞	4	3	7	2
	8	∞	6	1	2
	4	9	∞	2	3
	2	3	8	∞	4
	4	5	1	9	∞

23)	∞	7	3	1	3
	6	∞	9	4	3
	9	3	∞	7	5
	4	5	2	∞	7
	1	4	6	3	∞

24)	∞	5	2	3	3
	3	∞	3	4	2
	1	2	∞	9	8
	2	3	2	∞	5
	4	5	4	2	∞