**Введение**

Большой класс прикладных задач оптимизации сводится к задачам целочисленного линейного программирования. Для решения этих задач широко применяются методы отсечения, предназначенные для решения общей целочисленной задачи, сопоставляя ей некоторую нецелочисленную задачу, по решению которой и позволяет найти решение.

Первый методов решения целочисленной задачи линейного программирования отсечением был предложен Гомори и получил название алгоритма Гомори.

**1. Постановка задачи**

Метод Гомори предназначен для решения целочисленных задач линейного программирования. При рассмотрении метода Гомори будем решать данную задачу в канонической форме.

**1.1 Каноническая форма**

Будем рассматривать каноническую задачу целочисленного программирования с *n* переменными и *m*условиями, дополненную условием целочисленности:

Где *c = (c1, c2, … , cn)*, *x = (x1, x2, … , xn)* - вектора размерности *n*, *<c, x>* - их скалярное произведение (), называемое так же целевой функцией, *A* - матрица размерности *n*´ *m*, *b* - вектор-столбец размерности *m*.

Условие целочисленности существенно осложняет задачу линейного программирования (1.1), (1.2). Так может случиться, что задача (1.1)-(1.3) обладает допустимыми (целочисленными) планами, целевая функция ограничена на допустимом множестве, однако ее максимум не достигается. Например в задаче:

не существует целочисленных решений, в то время как без этого условия решением служит любой вектор вида

.

В связи со сказанным, при обосновании численных алгоритмов решения задач типа (1.1)-(1.3) приходится накладывать различные дополнительные условия.. Будем считать, что множество *X* планов задачи (1.1), (1.2) (без условия целочисленности) ограничено, то есть является многогранником.

Из этого условия вытекает, что множество *X\** всех целочисленных планов задачи (1.1)-(1.3) конечно.. Будем предполагать, что все коэффициенты *cj*, *j=1*, *2*, …, *n*, целевые функции - целые числа.

Из условия II вытекает, что для всякого целочисленного плана *xÎX*\* значение <*c*, *x*> максимизируемой линейной формы - целое число. В этом случае говорят, что гарантирована целочисленность целевой функции.

Хотя условия I и II на задачу (1.1) - (1.3) довольно жесткие, ослабить их, для получения необходимых результатов, можно лишь немного.

**1.2 Приведение к канонической форме**

Задача целочисленного линейного программирования может формулироваться несколько иначе, нежели это было приведено выше. Так, например, вместо условия (1.2) может иметь место другая форма записи ограничений

От подобной записи к (1.2) можно перейти, прибавив к каждому уравнению по одной новой переменной, тогда ограничения примут вид

Добавленные переменные будут иметь нулевой вес в целевой функции.

Отметим, что для получения, в зависимости от вида неравенства, следует не только прибавлять, но и вычитать переменную в зависимости от знака неравенства, учитывая условие (1.3).

Если в исходной задаче не для некоторой переменной *xi* не выполнено условие положительности, то ее следует заменить на разность двух новых, положительных, переменных.

**2. Общие идеи методов отсечения**

Существует принципиальная возможность свести решение задачи (1.1) - (1.3) к нахождению оптимального плана некоторой задачи линейного программирования (без условия целочисленности переменных). Пусть *X* - многогранное множество, определяемое условиями (1.1), (1.2). Через *X\** обозначим множество всех целочисленных векторов *x\**, лежащих в *X*. Другими словами *X\** задается условиями (1.1)-(1.3), т.е.

По определению *X\**Í*X*. Будем обозначать через *Xz* выпуклую оболочку множества *X\**. Тогда *Xz*Í*X*.

Таким образом, мы сопоставили многогранному множеству X некоторое выпуклое множество *Xz*Í*X* согласно следующему правилу: *Xz* есть минимальное выпуклое множество, содержащие все целочисленные векторы из *X*.

По предположению I, *X* является многогранником, следовательно множество *X\** конечно. Тогда выпуклое множество *Xz* так же является многогранником, а следовательно, имеем, что *Xz* можно задать конечным числом линейных неравенств.

Чтобы подчеркнуть основное отличие множества *Xz* от множества *X*, дадим следующее определение.

**Определение 1**. Многогранник, все крайние точки которого целочисленны (т.е. все их координаты целые числа), называются целочисленным многогранником.

Очевидно, что многогранник *Xz* - целочисленный, по скольку его крайними точками являются лишь точки множества *X\** целочисленных планов.

Оправданием для изучения соответствия *X®Xz* является следующий простой факт.

**Теорема 1**. Любой оптимальный опорный план задачи линейного программирования является решением задачи (1.1)-(1.3).

**Доказательство.** Пусть `*x\** - оптимальный опорный план задачи (2.1), *x\** - какое то решение исходной задаячи (1.1)-(1.3). `*x\**Î*Xz*Í*X*, то

<*c*,`*x\**> £ <*c*, *x\**>.

С другой стороны, так как *x\** - целочисленный план, то *x\**Î*X\**Í*Xz*, и поэтому

<*c*,`*x\**> ³ <*c*, *x\**>,

откуда

<*c*,`*x\**> = <*c*, *x\**>.

Доказательство теоремы закончено.

Подчеркнем, что теорема 1 утверждает лишь принципиальную возможность сведения решения задачи целочисленного линейного программирования к поиску опорных планов задачи линейного программирования вида (2.1). Основная трудность в использовании этой возможности состоит в явном задании многогранника *Xz*системой линейных неравенств с тем, что бы затем применить для решения задачи (2.1) численные методы линейного программирования. Вероятнее всего, что в вычислительном отношении эта проблема столь же сложна, как и исходная задача поиска оптимального целочисленного плана.

Несмотря на это, идея «сужения» множества *X* оказалась полезной и привела к созданию нескольких алгоритмов решения целочисленных задач линейного программирования, носящих название «методы отсечения».

Изложим идею методов отсечения. Допустим, что удалось построить последовательность {*Lr*}, *r = 0*, *1*, *2*, …, задач линейного программирования, каждая из которых определяется своим многогранником *Xr* и одной для всех целевой функцией <*c*, *x*>:

при чем эта последовательность задач обладает следующими свойствами:

) *X0=X*, т.е. в качестве *X0* берется множество планов задачи (1.1),(1.2);

) *Xrz = Xz, r=1,2, …*;

) если при решении задачи *Lr* полученный оптимальный вектор *xr\** не удовлетворяет условию целочисленности, то он не является планом задачи *Lr+1*, т.е. *xr\**Ï*Xr+1*.

Отметим, что если на каком то шаге *r* вектор *xr\** - решение задачи *Lr* - оказался целочисленным, то он является решением исходной задачи (1.1)-(1.3) в силу свойства 2) последовательности *Lr*.

Интуитивно ясно, что последовательное построение задач *Lr*, *r=1,2,* …, дает в некотором смысле аппроксимацию множества *Xz* с помощью множества *Xr*.

Способы построения последовательности {*Lr*}, обеспечивают конечность процесса решения задачи (1.1)-(1.3), были впервые предложены Гомори.

**3. Описание метода Гомори**

Рассмотрим теперь алгоритм Гомори для решения целочисленных задач линейного программирования. Этот метод принадлежит к числу методов отсечения и реализует идеи, изложенный в предыдущем пункте.

**3.1 Понятие правильного отсечения и простейший способ его построения**

алгоритм гомора линейное программирование

Введем правильного отсечения, которое лежит в основе многих численных методов решения целочисленных задач линейного программирования.

**Определение 2**. Пусть x\* - оптимальный план задачи (1.1), (1.2), не являющийся целочисленным. Неравенство

где g = (g*1*, g*2*, …, g*n*), называется правильным отсечением, если оно удовлетворяет требованиям:

а) для вектора *x\** неравенство не выполняется, т.е. <g, *x\**> > g*0* (условие отсечения).

б) если *x* - целочисленный план задачи (1.1), (1.2) ( т.е. план задачи (1.1)-(1.3)), то *x* - удовлетворяет (3.1) (условие правильности).

Понятно, что добавление неравенства (3.1) к условиям (1.1), (1.2) сужает допустимое множество *X*, сохраняя при этом все его целочисленные точки. Тем самым последовательное применение этого приема дает как бы многоэтапную аппроксимацию многогранника *Xz* с помощью линейных ограничений.

В связи с понятием правильного отсечения возникают две проблемы. Первая из них состоит в том, что бы сформулировать общий алгоритм отсечения для произвольной целочисленной задачи линейного программирования. Вторая проблема состоит в построении правильного отсечения таким образом, что бы обеспечить решение задачи (1.1)-(1.3) за конечное число шагов, т.е. чтобы от множества *X* всякий раз отсекались достаточно большие участки.

Отметим, что второе требование является довольно тонким. В качестве подтверждения рассмотрим способ построения правильного отсечения, предложенный Данцигом.

Пусть *x\** - опорный оптимальный план задачи (1.1), (1.2), s и w - списки номеров соответственно базисных и не базисных переменных, отвечающих некоторому базису плана *x\**. Тогда *xj\** = 0 при *j*Îw. С учетом этого свойства нетрудно построить правильное отсечение для плана x\*, если он является не целочисленным: в качестве такого может служить неравенство

.

В самом деле, условие отсечения тривиально выполняется, поскольку . Условие правильности так же соблюдено, так как если *x =* (*x1*,*x2* , …,*xn*) - целочисленный план задачи (1.1), (1.2), то величина с учетом условий *xj* ³ 0, *j*Î*w*, может быть меньше единицы лишь в том случае, когда *xj* = 0 при всех *j*Î*w*. Но в таком случае план *x* - опорный, и в качестве его базиса можно принять базис плана *x\**. Опорный план однозначно определяется своим базисом, откуда получаем, что из неравенства вытекает *x=x\**. Последнее, однако, невозможно, так как *x* - целочисленный вектор, а *x\** таковым не является.

Указанный прием построения правильного отсечения, несмотря на его простоту, оказался малоэффективным, поскольку конечность процесса решения обеспечивается лишь для узкого класса целочисленных задач линейного программирования.

**3.2 Построение правильного отсечения методом Гомори**

Опишем способ построения правильного отсечения, предложенный Гомори. Для произвольного числа *a*, через [*a*] будем обозначать его целую часть, т.е. [*a*] есть наибольшее целое число *k* непревосходящее число *a*.

Дробной частью {*a*} числа *a* называется число {*a*} = *a*- [*a*]. Отметим очевидное свойство дробной части произвольного числа: 0£{*a*}<1, причем {*a*} = 0 в том и только в том случае, когда *a*- целое число.

Пусть *x\** - опорное решение задачи (1.1), (1.2), не являющееся целочисленным, - его базис и *B* - соответствующая симплекс-таблица в координатной форме.

Рассмотрим приведенную систему уравнений, отвечающую данному базису (и таблице *B*) плана

*x\**:

Поскольку *xj\** = 0 при *j*Îw, то нецелочисленными могут быть лишь величины *x0\** = <*c*, *x\**>, *xi\**, *i*Îs.

Пусть *s* - такой индекс ( 0 £ *s* £ *n* ), что число *xs\** - не целое. Рассмотрим *s*-ю строку в симплексной таблице *B* (*s*-е уравнение в системе (3.2), (3.3)) и составим выражение

**Теорема 2**. Если *x*Î*X\** - целочисленный план задачи (1.1), (1.2), то

**Доказательство.** Используя соотношение

*asj* = [*asj*] + {*asj* }, *j* = 0, 1, 2, … , *n*, из (3.3) при *i*= *s* получаем

откуда

В левой части данного неравенства стоит целое число, следовательно, число

также целое. Из того, что *xj* ³ 0, *j*Îw, используя свойство дробной части, получаем

т.е. - *zs*(*x*) < 1, или *zs*(*x*) > -1. Учитывая, что *zs*(*x*) - целое, окончательно примем *zs*(*x*) ³ 0.

**Следствие.** Если *xs\**( = *as0* ) - нецелое число и Множество *X\** планов целочисленной задачи (1.1)-(1.3) непусто, то среди чисел {*asj*}, *j*=1, 2, …, *n*, есть положительные.

В самом деле, все числа {*asj*}, очевидно неотрицательны. Допустим, что {*asj*} = 0, *j* = 1, 2, …, *n*.

Если *X\** непусто и *x\** Î *X\**, то *zs*(*x\**)= - {*as0*}, о том, что *zs*(*x\**) - целое число, так как 0 < {*as0*} < 1.

Замечание. В доказательстве теоремы 2 мы воспользовали предположением II о том, что гарантирована целочисленность целевой функции. Действительно в случае *s* = 0 величина

является целой ( при условии, что *x*Î *X\**) лишь тогда когда число *x0* = < *c*, *x* > - целое.

Отсюда вытекает

**Теорема 3.** Если число *xs\** - нецелое, то неравенство

является правильным отсечением.

**Доказательство.** Проверим условие отсечения в определении 2. Так как *xs\** = *as0*, то из того, что *xs\** - нецелое, получаем неравенство {*as0*} > 0. Поскольку *xj\** = 0 при *j*Î w, то

,

и поэтому вектор *x\** не удовлетворяет неравенству (3.5). Условие правильности следует из утверждения *zs*(*x*) ³ 0 в теореме 2.

**3.3 Первый алгоритм Гомори**

Перейдем к изложению первого алгоритма Гомори.

Обозначим задачу (1.1), (1.2) через *L*0. Гомори предложил на первом этапе своего алгоритма находить лексикографический максимум задачи *L*0. Будем обозначать через

*x*(0) = (*x*0(0), *x*1(0), …, *x*n(0))

(n+1)-мерный вектор такой, что (*x*1(0), *x*2(0), …, *x*n(0)) - решение лексикографической задачи *L*0, а *x*0(0) = - значение линейной формы. В тех случаях когда это не вызывает недоразумения, будем называть *x*(0) - оптимальным планом лексикографической задачи *L*0 (хотя по общепринятой терминологии планом называется вектор, составленный из последних *n* координат вектора *x*(0)).

Отметим также, что *x*(0) будет являться опорным планом, а так же строго допустимым псевдопланом задачи*L*0.

Если x(0) - целочисленный вектор, то он, очевидно, и является решением задачи (1.1) - (1.3).

В противном случае отыскивается минимальный индекс s, 0 £ s £ n, для которого величина *xs*(0) не является целой. Пусть *B*(0) - симплексная таблица в координатной форме, соответствующая вектору *x*(0). С помощью коэффициентов *s*-й строки этой таблицы (т.е. коэффициентов приведенной системы (3.2), (3.3)) приемом, описанным выше, строится правильное отсечение.

Вводится вспомогательная переменная *xn*+1 и рассматривается задача *L*1:

Следующий этап состоит в нахождении лексикографического максимума задачи *L*1. Важным достоинством алгоритма Гомори является тот факт, что начальный допустимый строгий псевдоплан для применения двойственного симплекс-метода к задаче *L*1 находится без труда. Действительно, легко видеть, что в качестве такого псевдоплана можно взять вектор

где

В самом деле, очевидно, что *y*(1) удовлетворяет ( вместе с вектоорм *x*(0) ) ограничениям (3.6), (3.7) задачи *L*1, а из ограничений (3.8) нарушается лишь одно: *xn*+1(0)= - {a*s*0} < 0. Кроме того, *y*(1) является опорным для системы уравнений (3.6), (3.7), поскольку, если - базис плана *x*(0) то система

линейно независима и служит базисом для *y*(1). Покажем, что *y*(1) - строго допустимый псевдоплан. С этой целью построим симплексную таблицу, соответствующую указанному базису вектора *y*(1). Для этого нужно лишь приписать снизу к таблице *B*(0) строку

Где w = {*j*1, *j*2, …, *jn*-*m*} - список номеров небазисных переменных, соответствующих таблице *B*(0) опорного плана*x*(0). Поскольку *x*(0) - строго допустимый псевдоплан, то всякий столбец b*j*, *j*Îw, таблицы *B*(0) лексикографически положителен: b*j* > 0, *j*Îw. Отсюда вытекает, что и в симплексной таблице в координатной форме, отвечающей опорному вектору *y*(1), всякий столбец (кроме первого, совпадающего с *y*(1)) лексикографически положителен:

Таким образом, имея в своем распоряжении решение *x*(0) лексикографической задачи *L*0 и соответствующую симплекс таблицу в координатной форме *B*(0), без каких либо дополнительных вычислений находим начальный строго допустимый псевдоплан *y*(1) для задачи *L*1 и строим соответствующую ему симплексную таблицу в координатной форме.

Может случиться, что лексикографическая задача *L*1 не имеет решения. В этом случае решение целочисленной задачи (1.1) - (1.3) следует прекратить поскольку имеет место

**Теорема 4.** Если в задаче *L*1 не существует лексикографического максимума, то множество *X*\* целочисленных точек задачи (1.1) - (1.3) пусто.

**Доказательство.** Поскольку множество *X* векторов, удовлетворяющих условиям *Ax* = *b*, *x* ³ 0, согласно предположению I ограничено, то ограниченным является и множество планов задачи *L*1. Поэтому единственной причиной, по которой эта задача может не иметь лексикографического минимума, может быть только то что множество ее планов пусто. Покажем что в таком случае множество *X*\* также пусто.

Предположим противное, т.е. что *X*\* ¹ Æ, и пусть *x*\* = (*x*1\*, *x*2\*, …, *xn*\*) Î X\*. По теореме 2 получаем, что величина

неотрицательна. Но это означает, что вектор = (*x*1\*, *x*2\*, …, *xn*\*, *xn*+1\*) является планом задачи *L*1, в противоречие с вышесказанным. Теорема доказана.

Пусть *x*(1) = (*x*0(1), *x*1(1), …, *xn*(1), *xn*+1(1)) - решение лексикографической задачи *L*1. Отправляясь от задачи *L*1и вектора *x*(1), аналогичным образом строятся задачи *Lr*, *r* = 2, 3, …, и решения *x*(*r*) Î Â*n*+1+*r* соответствующим им лексикографическим задач.

Заметим, что алгоритм Гомори однозначно определяет последовательность *x*(*r*), *r* = 0, 1, 2, …, что следует из однозначности выбора *s*. Обратим так же внимание на то, что индекс *s* всегда не превосходит n: 0 s n. В самом деле, если все *xj*(*r*) при *j* = 0, 1, 2, …, n - целые числа, то из теоремы 2 вытекает, что *xn*+1(*r*), *xn*+2(*r*), … - также целые.

Если на каком - то шаге *r* вектор

*x*(*r*) = ( *x*0(*r*), *x*1(*r*), …, *xn*(*r*), …, *xn*+*r*(*r*) )

оказывается целочисленным, то вектор (*x*0(*r*), *x*1(*r*), …, *xn*(*r*)) является решением задачи (1.1) - (1.3). Доказательство этого утверждения очевидно.

Переход от вектора *x*(*r*) к вектору *x*(*r*+1) с помощью описанного алгоритма Гомори называется большой итерацией, в отличие от промежуточных итераций двойственного симплекс-метода, которые называются малыми.

**3.4 Доказательство конечности алгоритма**

Основной вопрос, относящийся к первому алгоритму Гомори таков: всегда ли за конечное число больших итераций можно получить целочисленный вектор *x*(*r*).

Докажем конечность первого Алгоритма Гомори. Будем использовать следующие обозначения:

*x*(0) = ( *x*0(0), *x*1(0), …, *xn*(0) );

где ( *x*1(0), …, *xn*(0) ) - решение лексикографической задачи L0, а *x*(0) = - соответствующее значение линейной формы (целевой функции).

*y*(1) = ( *x*0(0), *x*1(0), …, *xn*(0), *xn*+1 ),

где

вектор y(1) служит начальным строго довпустимым псевдопланом при решении задачи L1 двойственным симплексным методом в координатной форме: `*y*(1) = (`*y*0(1), `*y*1(1), …, `*yn*(1), `*yn*+1(1) ) - вектор, получающийся из *y*(1) в результате первой (малой) итерации двойственного симплекс метода в координатной форме.

Аналогично вводятся обозначения

*x*(*r*), *y*(*r* + 1), `*y*(*r* + 1), *r* = 1, 2, …

Из свойств двойственного симплекс метода в координатной форме следует

*y*(*r*) >`*y*(*r*) ³ *x*(*r*).

**Лемма 1.** Пусть *s* - минимальный индекс, для которого число xs(0) - не целое. Тогда

.

**Доказательство.** Поскольку из (3.9) следует *y*(1) >`*y*(1), то возможно два случая:

1..

2..

В случае 1 утверждение леммы выполняется тривиально по определению лексикографического порядка.

Рассмотри случай 2. Согласно правилам двойственного симплекс-метода на первой (малой) итерации решения задачи *L*1 выводу из числа базисных подлежит переменная *xn*+1, поскольку в векторе *y*(1) *xj*(0) ³ 0, *j* Î w, *xn*+1 < 0. Пусть *k* Î w - такой индекс, что

Для любого *j* Î w, если -{a*sj*} < 0. По правилам симплексного метода в число базисных вводится переменная *xk*.

Случай 2 возможен лишь при условии b*ik* = 0, *i* = 0, 1, 2, …, *s* - 1. Поскольку *x*(0) - строго допустимый псевдоплан задачи *L*0, то все ее столбцы b*j*, *j* Î w, симплекс таблицы *B*(0) лексикографически положительны; в частности b*k* > 0 . Следовательно, координата b*sk* данного столбца должна быть неотрицательной. Заметим, что b*sk* = a*sk* ( т.е. 0£ *s* £ *n* и *s* Î w ), по условию (3.10) число a*sk* - не нуль. Поэтому

a*sk* > 0 и a*sk* ³ {a*sk*}

По формуле преобразования симплекс-таблицы имеем

Вспоминая, что xs(0) = as0, получаем:

.

С учетом (3.11) получим оценку:

.

Лемма доказана.

**Замечание.** Если исходная задача (1.1) - (1.3) допустима, то согласно следствию из теоремы 2 индекс k, удовлетворяющий условию (3.10), существует.

**Следствие.**Справедливо соотношение

Действительно при *r* = 1 это неравенство вытекает из леммы и второго неравенства (3.9) . Что бы получить это утверждение при произвольном *r*, нужно воспользоваться тем, что *yj*(*r*) = *xj*(*r*) при 0 £ *j* £ *n*, и неравенством *y*(*r*) ³*x*(*r*) в (3.9).

**Теорема 5**. Если выполнены предположения I и II, то первый алгоритм Гомори требует лишь конечного числа больших итераций.

Что бы убедиться в истинности теоремы, необходимо показать, что при некотором *r* вектор *x*(*r*) = (*x*0(*r*), *x*1(*r*), …,*xn*+*r*(*r*)) - целочисленный. Для этого, в свою очередь, достаточно доказать целочисленность вектора (*x*0(*r*), *x*1(*r*), …, *xn*(*r*)), поскольку из теоремы 2 тогда вытекает, что все числа *xn*+1(*r*), *xn*+2(*r*), …, *xn*+*r*(*r*) также целые. Вспомним также, что минимальный индекс s, при котором число xs(r) - нецелое , всегда не превосходит n: 0 £ s £ n. Прежде чем переходить к основному доказательству докажем следующую лемму:

**Лемма 2.** Для любого *j*, 0 £ *j* £ *n*, существует такой номер *Rj*, что при *r* ³ *Rj* все числа *xj*(*r*) - целые и равны одному и тому же целому числу *xj*(*Rj*).

**Доказательство.** Пусть *s*, 0 £ *s* £ *n*, - минимальный индекс для которого утверждение Леммы не выполняется. Обозначим

В том случае когда *s* = 0, положим *R* = 0.

Пусть *r*, *l* - такие индексы, что *R* £ *r* £ l, и числа *xs*(*r*), *xs*(*l*) - нецелые. Покажем, что тогда [*xs*(*r*)] > [*xs*(*l*)]. Действительно по определению *s* имеем

.

В таком случае *s* - минимальный индекс, для которого число *xs*(*r*) - нецелое. По следствию из леммы 1 имеем [*xs*(*r*)] ³ *xs*(*l*).

Учитывая, что *xs*(*l*) - не целое число, имеем *xs*(*l*) > [*xs*(*l*)], откуда и получаем нужное утверждение. Поскольку множество *X* планов задачи (1.1) - (1.3) ограничено, то ограничена любая величина *xs*(*r*), 0 £ *s* £ *n*, *r* = 1, 2, … . Поэтому бесконечной цепочки неравенств вида [*xs*(*r*)] > [*xs*(*l*)] > … существовать не может, а, следовательно, в последовательности *xs*(*r*), *r* = 0, 1, …, не может быть бесконечно много нецелых чисел. Аналогично доказывается, что в этой последовательности не может быть бесконечно много различных целых чисел.

Лемма доказана.

Вернемся к доказательству теоремы. Пусть

,

где числа *Rj* фигурируют в формулировке леммы 2. Тогда согласно этой лемме все числа *xj*(*R*), 0 £ *j* £ *n*, - целые. Из теоремы 2 получаем, что вектор *x*(*R*) - целочисленный. Следовательно алгоритм Гомори требует не более *R*итераций.

Теорема доказана.

**3.5 Замечания по практической реализации первого алгоритма Гомори**

При практической реализации первого алгоритма Гомори важной проблемой является нарастание количества ограничений, что ведет к увеличению размеров симплексных таблиц. Гомори предложил способ устранения этого недостатка алгоритма, заключающийся в следующем.

) В ходе решения задачи *Lr* двойственным симплексным методом на каждой малой итерации следует пользоваться уточненным правилом вывода из числа базисных векторов для решения задач линейного программирования симплекс-методом: если в первом столбце симплексной таблицы имеется несколько отрицательных элементов b*i*0 ( = *xi*), *i* =1, 2, …, *n*, …, *n* + *r*, то выводить из числа базисных надо переменную с минимальным номером.

) Если в ходе очередной малой итерации при реализации задачи *Lr* все основные переменные *x*1, *x*2, …, *xn*оказались неотрицательными, то дальнейшее применение двойственного симплекс-метода к задаче *Lr* следует прекратить, несмотря на то, что ее лексикографический максимум, быть может, еще не достигнут. Если при этом все переменные *xj*, *j* = 1, 2, …, *n*, оказались целочисленными, то по теореме 2 все вспомогательные переменные*xn*+*k*, *k* = 1, 2, …, *r*, целочисленны и неотрицательны. Это означает, как уже известно, что вектор ( *x*0, *x*1, *x*2, … , *xn*) является решением исходной целочисленной задачи. В противном случае переходим к новой большой итерации.

) Строка симплексной таблицы, соответствующая вспомогательной переменной *xn*+*r*, удаляется, как только переменная *xn*+*r* объявляется небазисной. Напомним, что это происходит на первой же малой итерации решения задачи *Lr*.

) Если в ходе решения задачи *Lr* переменная *xn*+*r* вновь попадает в число базисных, то то соответствующая ей строка не восстанавливается.

Понятно, что при выполнении правил 3), 4) размеры симплексной таблицы в первом алгоритме Гомори не увеличиваются - в каждой таблице содержится *n* + 2 строк, отвечающие основным переменным *x*0, *x*1, … , *xn* и текущей вспомогательной переменной *xn*+*r* в момент ее введения) и *n* - *m* +1 столбцов ( поскольку число *n* - *m*небазисных переменных не меняется ).

) На первой малой итерации решения задачи *Lr*+1 в качестве переменной, выводимой из базиса, выбирается именно *xn*+*r*+1, не смотря на то, что значения остальных вспомогательных переменных в этот момент так же могут быть отрицательными.

Заметим, что правило 5) на самом деле избыточно, поскольку при выполнении правил 3) и 4) мы ничего не знаем о значении остальных переменных *xn*+1, …, *xn*+*r* в момент перехода к задаче *Lr*+1. Данное правило выделено лишь для того, чтобы подчеркнуть отличие рассматриваемых алгоритмов.

Отметим, что при использовании правила 2) возникающая последовательность *x*(*r*) может не быть лексикографическим максимумом задачи *Lr*, поскольку значения некоторых из вспомогательных переменных могут быть отрицательными.

Тем не менее, для последовательности *x*(*r*), *r* = 0, 1, 2, …, получаемой с использованием правил 1) и 2), сохраняется важное свойство: эта последовательность единственна.

Осталось лишь доказать, что при использовании правил 1) - 4) алгоритм Гомори остается конечным, поскольку его конечность и будет означать, что он приводит к цели - нахождению целочисленного решения задачи (1.1) - (1.3). В самом деле, конечность числа *R* больших итераций означает, что вектор *x*(*R*) целочисленный.

Отметим, во-первых, что при использовании правила 2) число малых итераций решения задачи *Lr* конечно - не больше, чем требуется для нахождения ее лексикографического максимума.

Теорема 6. Последовательность x(r), возникающая в процессе применения алгоритма Гомори, уточненного правила 1) - 4), конечна.

Доказательство. Заметим, что в доказательстве теоремы 5 о конечности последовательности x(r) использовались лишь два обстоятельства, регулирующие возникновение этой последовательности: способ построения правильного отсечения и тот факт, что во всякой текущей симплекс-таблице вс столбцы b*j*, *j*Îw, лексикографически положительны. Таким образом, удаление строки, соответствующей вспомогательной переменной, может повлиять лишь на последнее обстоятельство. Этого, однако, так же быть не может, как показано в следующей лемме.

**Лемма 3.** В любой симплекс-таблице, возникающей в ходе алгоритма Гомори, уточненного правилами 1) - 4), для любого столбца

имеет место неравенство

.

**Доказательство.** Допустим, что утверждение леммы не выполняется для некоторого *k* Î w. Поскольку b*k*, то данное предположение означает, что

По определению симплексной таблицы в координатной форме, имеем

Для любого *x* Î *Rn*+1+*r* , если утверждение леммы нарушается в ходе решения задачи *Lr*. Формула (3.13) с учетом (3.12) означает, что изменение значения переменной *xk* не влияет на значение *xi*, *i* = 0, 1, 2, …, *n*. Другими словами, при одном и том же наборе величин *xi*, 0£*i*£*n*, переменная *xk* может принимать произвольное значение. Отсюда следует, во-первых, что *k* ³ *n* + 1, а во-вторых, что принятое допущение (3.13) неверно, поскольку поскольку значение любой вспомогательной переменной *xk*, *k* ³ *n* + 1, как вытекает из (3.7), однозначно определяется значениями основных переменных.

Поскольку удаление строк, соответствующих вспомогательных переменным, не влияет на свойство столбцов b*j*, *j*Î w, быть лексикографически положительными, то эти строки вообще не нужны. Действительно, с учетом правил 1) - 2) переменная *xn*+*r*, попав в число базисных, так и остается базисной до конца вычислений, и ее строка не потребуется для определения переменной, вводимой в базис согласно правилам симплекс-метода.

Таким образом, элементы строки, соответствующие переменной *xn*+*r*, не участвуют в формулах двойственного симплекс-метода для вычисления значений всех остальных переменных.

Поскольку, как отмечалось, индекс *s*, регулирующий формирование правильного отсечения, не превосходит *n*, 0£ *s* £ *n*, то и для этих целей вспомогательные переменные не потребуются.

**4. Реализация на ЭВМ**

В данном курсовом проекте программа предназначена для нахождения решения целочисленной задачи линейного программирования методом Гомори.

В программном модуле используется алгоритм описанный в теоретической части с учетом замечаний по практическому применению метода, сделанных Гомори.

Ввод данных в программном модуле производится из файла. Вывод результатов работы программы может производится в файл, на дисплей или на принтер. Формат входного файла:

<m> <n>

<c1> <c2> … <cn>

<a11> <a21> … <an1> <b1>

<a12> <a22> … <an2> <b2>

…

<a1m> <a2m> … <anm> <bm>

где *n* - количество переменных, *m* - количество ограничений, *c*1, *c*2, … , *cn* - коэффициенты максимизируемой линейной формы, *aij* - элементы матрицы *A, bj* - компоненты вектора *b*. A, b - характеризуют ограничения [см. (1.2)].

Пример входного файла:

9

2 5 0 0 0 0 0 0 0

3 1 0 0 0 0 0 0 12

2 5 0 1 0 0 0 0 0 30

3 2 0 0 1 0 0 0 0 22

2 -1 0 0 0 1 0 0 0 12

1 -3 0 0 0 0 1 0 0 0

2 5 0 0 0 0 0 -1 0 10

5 1 0 0 0 0 0 0 -1 5

**Список литературы:**

1. Абрамов Л.А., Капустин В.Ф. Математическое программирование. - Л.: Изд-во ЛГУ, 1981. -328 с.

. Белоусова Г.С. Линейное программирование. Учебное пособие. -Красноярск: Наука, 1975. -107 с.

. Кузнецов Ю.Н. и др. Математическое программирование: Учебное пособие. - 2-е изд., перераб. и доп. -М.: Высшая школа, 1980. -300 с.

. Ашманов С.А., Линейное программирование. М.: Наука. 1969. -240 с

. Габасов Р.И. Кириллова Ф.М., Методы линейного программирования. Минск: Наука. 1977. -174 с