Оглавление

Дискретное программирование сформировалось как самостоятельная и важная часть математического программирования в конце 60-х годов. Основная задача дискретного программирования – выбор наилучшего варианта из конечного, возможно, очень большого их числа. Возникающие при этом экстремальные задачи имеют ряд особенностей, которые не встречаются в таких стандартных задачах математического программирования, как линейные, выпуклые или многокритериальные задачи.

1. Постановка и особенности задач дискретного программирования

Под *задачей дискретного программирования* (дискретной оптимизации) понимается задача математического программирования

$$f\left(x^0\right) = \min f\left(x\right), x \in G, \tag{1.1}$$

множество допустимых решений которой конечно, т.е. $0 \le |G| = N < \infty$, где $|G|$ - число элементов множества G. В силу конечности G все допустимые решения можно пронумеровать:

$$x^1, x^2, \ldots, x^N,$$

вычислить $f\left(x^i\right), i = 1,2,\ldots,N$, и найти наименьшее значение. Однако такой метод полного перебора при решении задачи реализовать невозможно, так как N может оказаться настолько большим, что этот перебор невыполним на ЭВМ любой мощности.

Во многих задачах условия дискретности отделены от других условий, т.е. если $x = \left(x_1, x_2, \ldots, x_n\right)$, то $x_j \in G_j = \left(x_1^j, x_2^j, \ldots, x_{kj}^j\right)$, $k_j \ge 2$. Поэтому $N = \prod_{j=1}^{n} |G| = \prod_{j=1}^{n} k_j \ge 2^n$, отсюда видно, что с ростом числа переменных объем вычислительной работы резко возрастает.

В качестве примера задачи дискретного программирования рассмотрим задачу частично целочисленного линейного программирования – одну из наиболее часто встречающихся в приложении и наиболее изученных задач.

Задача формулируется следующим образом:

$$\sum_{j=1}^{n} c_j x_j \to \min, \tag{1.2}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1,2,\ldots,n, \tag{1.3}$$

$$x_j \ge 0, \quad j = 1,2,\ldots,n \tag{1.4}$$

$$x_j - \textit{целые}, \quad j = 1,2,\ldots n_1 \le n. \tag{1.5}$$

Если $n_1 < n$, то задача называется *частично целочисленной*, если $n_1 = n$ - полностью целочисленной. Среди задач рассматриваемого класса выделяются задачи с булевыми переменными

$$x_j \in \{0,1\},\ j = 1,2,\ldots,n_1 \le n.$$

Особенности задач

*Нерегулярность***.** Дискретные задачи математического программирования входят в класс нерегулярных задач. Регулярные задачи математического программирования характеризуются следующими условиями:

- для каждой точки $x \in G$ можно определить некоторым образом понятие непустой окрестности $G_x \subset G$;
- можно указать достаточно эффективно проверяемые необходимые и достаточные условия локальной оптимальности; на основании этих условий локальный оптимум целевой функции на G может быть найден при помощи конечного (или бесконечного) сходящегося процесса;
- локальный оптимум целевой функции совпадает с глобальным.

К регулярным относятся задачи линейного и выпуклого программирования. К нерегулярным, наряду с дискретными, относятся многоэкстремальные задачи, в которых локальный экстремум может не совпадать с глобальным. Дискретные задачи характеризуются тем, что область допустимых решений невыпукла и несвязна, а это делает невозможным решение их с помощью стандартных методов, применяемых для решения регулярных задач математического программирования.

Трудности при определении окрестности. В задачах регулярного математического программирования значительная часть методов основана на следующем исходном положении: если точки $x^1 \in G$ и $x^2 \in G$ близки, то значения $f\left(x^1\right)$ и $f\left(x^2\right)$ также близки. В задачах дискретной оптимизации это положение не имеет места. Поэтому близость двух точек может быть оценена лишь по значениям функции в них.

Например, под *окрестностью булевого вектора* $x = \left(x_1, x_2, ..., x_n\right)$ будем понимать все векторы, получаемые из x заменой любой единицы нулем или любого нуля единицей. Рассмотрим задачу с тремя переменными:

$$x_1 + C_2 x_2 + C_3 x_3 \to \max,$$

$$x_1 + x_2 + x_3 \le 2,$$

$$1 < C_2 < C_3,$$

$$x_j \in \{0,1\},\ j = 1,2,3.$$

Оптимальное решение очевидно: $x = (0;1;1),\ f\left(x^0\right) = C_2 + C_3$. Вектор $x^1 = (0;1;0)$ находится в окрестности вектора x^0, $f\left(x^1\right) = C_2$. Ясно, что разность $f\left(x^0\right) - f\left(x^1\right) = C_3$ может быть сделана сколь угодно большой при помощи выбора величины C_3.

Трудности нахождения допустимого целочисленного плана в задаче (1.2)-(1.5). Предположим, что рассматривается задача частично целочисленного линейного программирования общего вида. Вопрос о существовании допустимого решения сводится к выяснению, разрешима ли система линейных равенств и неравенств в целых неотрицательных числах. Известно, что это трудоемкая вычислительная задача. Поэтому в общей задаче рассматриваемого класса поиск допустимого решения может оказаться столь же трудоемким, как и поиск оптимального решения.

Невозможность выполнения большого перебора на ЭВМ. Многие из методов решения задач дискретной оптимизации основаны на идее перебора вариантов, качество алгоритмов оценивается числом точек x, для которых вычислялись значения $f(x)$ или значения некоторых других функций. Объем вычислительной работы резко возрастает с ростом числа переменных. Поэтому перебор большого числа вариантов принципиально невозможен ни при каком быстродействии ЭВМ.

2. Методы отсечения

Выделим следующие (основные) методы решения задач дискретного программирования:

- метод отсечения для задач (частично) целочисленного линейного программирования;
- комбинаторные методы;
- приближенные методы.

Первые попытки решения задач целочисленного линейного программирования сводились к отбрасыванию условия целочисленности, решению линейной задачи и округлению полученного решения. Можно привести простой пример, который показывает неприемлемость такого подхода:

$$x_1 - 3x_2 + 3x_3 \to \max,$$

$$2x_1 + x_2 - x_3 \le 4,$$

$$4x_1 - 3x_2 \le 2,$$

$$-3x_1 + 2x_2 + x_3 \le 3,$$

$$x_1, x_2, x_3 \ge 0 \ \textit{целые}.$$

Игнорируя условие целочисленности, получим оптимальное решение линейной задачи $x_1 = \frac{1}{2}, x_2 = 0, x_3 = 4\frac{1}{2}$. Никакие округления компонент этого решения не дают даже целочисленного плана. Оптимальное решение целочисленной задачи имеет вид $x_1 = 2, x_2 = 2, x_3 = 5$.

Описанный подход основан на идее сведения решения нерегулярной задачи к решению конечной последовательности регулярных задач, неудача связана с «прямолинейным» применением идеи сведения. Применительно к задаче целочисленного линейного программирования идея регуляризации позволяет свести решение этой задачи к решению последовательности специальным образом построенных линейных задач. Она состоит в следующем. Если полученное решение удовлетворяет условию целочисленности, то процесс окончен. В противном случае к ограничениям исходной задачи добавляется новое линейное ограничение, обладающее двумя свойствами:

- полученный нецелочисленный план ему не удовлетворяет;
- любой целочисленный план ему удовлетворяет.

Решается задача с дополнительно введенным ограничением, процесс повторяется до получения целочисленного решения. Идея отсечения приводит к трем проблемам:

- нахождение универсального правила формирования отсечений;
- доказательство конечности процесса отсечения;
- борьба с чрезмерным «разрастанием» размерности задач при добавлении ограничений.

Только решение этих проблем может привести к универсальному и реализуемому в вычислительном отношении алгоритму. Р. Гомори удалось решить эти проблемы.

В дальнейшем выявилась большая непредсказуемость в поведении различных алгоритмов отсечения. Для некоторых вариантов алгоритма Гомори существуют задачи, в которых число отсечений быстро растет с ростом размерности задачи и ростом коэффициентов, а в текущих симплекс-таблицах появляются весьма большие числа.

Методы отсечения не нашли широкого применения при решении прикладных задач по следующим причинам:

- определение того, какое отсечение (из большого их числа) есть сильнейшее, есть сложная в вычислительном отношении задача;
- методы отсечения приспособлены в основном для решения чисто целочисленных задач, которые составляют лишь малую часть задач, встречающихся в приложениях;
- методы отсечения не приспособлены для решения задач со слабо заполненными матрицами.

Алгоритм Гомори

Приведем в качестве примера один из алгоритмов Гомори для решения полностью целочисленной задачи линейного программирования. Рассмотрим полностью целочисленную задачу линейного программирования:

$$\max(\min) Z = \sum_{j=1}^{n} c_j x_j ; \tag{2.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, i = 1,\dots,m , \tag{2.2}$$

$$x_j \ge 0, j = 1,\dots,n , \tag{2.3}$$

$$x_j - \text{целые}, j = 1,\dots,n . \tag{2.4}$$

Алгоритм Гомори состоит из следующих этапов.

1. Решается задача (2.1) – (2.3) с отброшенным условием целочисленности.

2. Полученное оптимальное решение (если оно существует) проверяется на целочисленность. Если условие целочисленности выполняется по всем переменным, то оптимальное решение задачи (2.1) – (2.4) совпадает с оптимальным решением задачи (2.1) – (2.3). Если это условие не выполняется хотя бы для одной переменной, то переходят к третьему этапу. Если задача (2.1) – (2.3) оказывается неразрешимой, то задача (2.1) – (2.4) тоже не имеет решения.

3. Строится дополнительное ограничение, отсекающее часть области, в которой содержится оптимальное решение задачи (2.1) – (2.3) и не содержится ни одного допустимого решения задачи (2.1) – (2.4).

4. Последний этап предусматривает возвращение к ЗЛП с отброшенным условием целочисленности, но с расширенной системой ограничений, в которую включено дополнительное ограничение, полученное на третьем шаге. К расширенной системе ограничений вновь применяется симплексная процедура. Если найденное таким образом решение будет опять нецелым, то формируется новое дополнительное ограничение и процесс вычислений повторяется.

Алгоритм Гомори позволяет за конечное число шагов прийти к оптимальному целочисленному решению, если оно существует.

С геометрической точки зрения каждому дополнительному линейному ограничению в n-мерном пространстве соответствует определенная гиперплоскость, отсекающая от многогранника решений некоторую его часть, включая и оптимальную на данном этапе нецелочисленную вершину. При этом все точки с целочисленными координатами, в том числе и искомая оптимальная, остаются в усеченном многограннике. Так как множество целых точек усеченного многогранника совпадает с множеством целых точек исходного многогранника, то понятно, что если оптимальное решение ЗЛП на усеченном многограннике удовлетворяет условию целочисленности, то оно является оптимальным целочисленным решением и исходной задачи. Через несколько операций отсечения искомая целочисленная точка оказывается сначала на границе, а затем становится оптимальной вершиной неоднократно усеченного многогранника решений.

Может оказаться, что многогранник решений не содержит ни одной целочисленной точки. В этом случае, сколько бы ни вводили дополнительных ограничений, целочисленного допустимого решения получить нельзя.

Основное в алгоритме – составление дополнительного ограничения, т.е. отсекающей гиперплоскости, которая называется *правильным отсечением.* Правильное отсечение должно удовлетворять следующим условиям: 1) быть линейным; 2) отсекать найденное оптимальное нецелочисленное решение задачи (2.1) – (2.3); 3) не отсекать ни одной из целочисленных точек задачи (2.1) – (2.4).

Рассмотрим, как можно сформировать правильное отсечение. После каждой итерации система ограничений имеет вид

$$x_i = b_i - \sum_{x_j \in \{СП\}} \alpha_{ij} x_j,\ x_i \in \{БП\}, \tag{2.5}$$

где $\{БП\}$ - множество базисных переменных.

Если выполняется условие оптимальности задачи, то находим оптимальное решение. Если все компоненты оптимального плана целочисленны, то задача решена. Предположим, что некоторые β_0 - нецелые. Пусть компонента i_0 - нецелая. Рассмотрим i_0- равенство системы (2.1.5), для которой выполняется условие оптимальности, т.е.

$$x_{i_0} = \beta_{i_0} - \sum_{x_j \in \{СП\}} \alpha_{i_0 j} x_j\ . \tag{2.6}$$

Найдем целую и дробную часть коэффициентов β_{i_0} и $\alpha_{i_0 j}$:

$$\beta_{i_0} = \lfloor \beta_{i_0} \rfloor + \{\beta_{i_0}\}, \tag{2.7}$$

$$\alpha_{i_0 j} = \lfloor \alpha_{i_0 j} \rfloor + \{\alpha_{i_0 j}\}, x_j \in \{СП\}\ .$$

Так как, по предположению, β_{i_0} не целое, то $\{\beta_{i_0}\} > 0$. Кроме того, $\{\alpha_{i_0 j}\} \geq 0$. Подставив выражение (2.7) в равенство (2.6), получим

$$x_{i_0} = ([\beta_{i_0}] - \sum_{x_j \in \{C\Pi\}} [\alpha_{i_0 j}] x_j) + (\{\beta_{i_0}\} - \sum_{x_j \in \{C\Pi\}} \{\alpha_{i_0 j}\} x_j). \qquad (2.8)$$

Так как первое слагаемое равенства (2.8) есть целое число, то, для того чтобы x_{i_0} было целым, необходимо, чтобы второе слагаемое также было целым числом, т.е. величина

$$L_{i_0} = \{\beta_{i_0}\} - \sum_{x_{j \in \{C\Pi\}}} \{\alpha_{i_0 j}\} x_j$$

должна быть целым числом. Так как x_{i_0} - координата допустимого целочисленного решения задачи (2.1) – (2.4), то L_{i_0} - всегда целое число. Легко показать, что $L_{i_0} \leq 0$. Таким образом, любое допустимое решение задачи (2.1) – (2.4) должно удовлетворять неравенству

$$\{\beta_i\} - \sum_{x_j \in \{C\Pi\}} \{\alpha_{i_0 j}\} x_j \leq 0. \qquad (2.9)$$

Справедлива теорема.

Неравенство (2.9) определяет правильное отсечение Гомори, т.е.:

1) является линейным;

2) отсекает найденное оптимальное нецелочисленное решение задачи (2.1)–(2.3);

3) не отсекает ни одного целочисленного плана задачи (2.1) – (2.4).

Если после очередной итерации окажется, что в оптимальном плане задачи (2.1) – (2.3) имеется несколько нецелых координат, то для построения отсекающей гиперплоскости целесообразно выбрать строку, содержащую свободный член с наибольшей дробной частью.

Признаком отсутствия целочисленности решения служит появление в симплексной таблице хотя бы одной строки с дробным свободным членом и целыми остальными коэффициентами, так как в этом случае соответствующее уравнение не имеет решения в целых числах.

Пример 2.1. Решить задачу

$$\max Z = x_1 + x_2;$$

$$-x_1 + 3x_2 \le 3,$$

$$3x_1 - x_2 \le 3,$$

$$x_1, x_2 \ge 0 - \textit{целые.}$$

Р е ш е н и е. В результате решения задачи без учета условия целочисленности переменных получаем оптимальный план $x_{нц}^{*(1)} = (3/2;3/2;0;0)$, содержащийся в таблице 2.1.

Таблица 2.1

БП	$c_Б$	A_0	x_1	x_2	x_3	x_4
			1	1	0	0
x_2	1	3/2	0	1	3/8	1/8
x_1	1	3/2	1	0	1/8	3/8
$z_j - c_j$		3	0	0	1/2	1/2

Таблица 2.2

БП	$c_Б$	A_0	x_1	x_2	x_3	x_4	x_5
			1	1	0	0	0
x_2	1	3/2	0	1	3/8	1/8	0
x_1	1	3/2	1	0	1/8	3/8	0
-	-	4	0	0	1	3	-1
$z_j - c_j$		3	0	0	1/2	1/2	0
x_2	1	4/3	0	1	1/3	0	1/24
x_1	1	1	1	0	0	0	1/8
x_4	0	4/3	0	0	1/3	1	-1/3
$z_j - c_j$		7/3	0	0	1/3	0	1/6

В полученном плане две компоненты нецелые. Поскольку они равны по величине, сформируем правильное отсечение (2.9), например, по

x_1-строке: $\{3/2\}-\{1/8\}x_3-\{3/8\}x_4 \ge 4$. Эквивалентное уравнение имеет вид $x_3+3x_4-x_5=4$. Дополняя табл. 2.1 строкой для этого уравнения и столбцом для новой переменной x_5, получаем расширенную задачу, записанную в табл. 2.2. В этой же таблице в результате симплексного преобразования найден оптимальный нецелочисленный план $x_{нц}^{*(2)}=(1;4/3;0;4/3;0)$ расширенной задачи. Продолжаем процедуру отсечения.

Новое правильное отсечение (2.9) сформируем, например, по x_2-строке: $\{4/3\}-\{1/3\}x_3-\{1/24\}x_5 \le 0$ или $8x_3+x_5 \ge 8$. Эквивалентное этому неравенству уравнение принимает вид $8x_3+x_5-x_6=8$.

Вновь расширяем задачу, добавляя строку для нового уравнения и столбец для переменной x_6. В результате табл. 2.2 преобразуется в табл. 2.3.

Таблица 2.3

БП	$c_Б$	A_0	x_1	x_2	x_3	x_4	x_5	x_6
			1	1	0	0	0	0
x_2	1	4/3	0	1	1/3	0	1/24	0
x_1	1	1	1	0	0	0	1/8	0
x_4	0	4/3	0	0	1/3	1	-1/3	0
-	-	8	0	0	[8]	0	1	-1
z_j-c_j		7/3	0	0	1/3	0	1/6	0
x_2	1	1						
x_1	1	1						
x_4	0	1						
x_3	0	1						
z_j-c_j		2	0	0	0	0	1/8	1/24

После очередной итерации получен оптимальный план $x_ц^{*(3)}=(1;1;1;1;0;0)$ новой расширенной задачи. Все компоненты этого плана – целые числа. Максимальное значение целевой функции выражается целым числом: $\max Zц=2$, а поэтому решение задачи закончено. Итак, для исходной задачи получили $x^*=(1,1)$, $Z^*=2$.

Упражнения для самостоятельной работы

2.2. Дать геометрическую интерпретацию приведенного решения задачи.

2.3. Решить задачи методом отсечения:

а) $f = x_1 + 2x_2 + 3x_3 (\max)$;

$$6x_1 + 4x_2 + 3x_3 \le 25,$$

$$5x_1 + 3x_2 + 2x_3 \le 15,$$

$$x_j \ge 0 (j = 1,2,3) - \textit{целые}.$$

б) $f = x_1 + 2x_2 + x_3 (\min)$;

$$x_1 + x_2 + 2x_3 \ge 3,$$

$$2x_1 + x_2 \ge 1,$$

$$x_j \ge 0 (j = 1,2,3) - \textit{целые}.$$

3. Задача о ранце

Задача об одномерном ранце. Пусть имеется n предметов, каждый из которых имеет ценность $c_j > 0$ и вес $a_j > 0,\ j = 1,2,...,n$. Имеется ранец (рюкзак), грузоподъемность которого R, при этом $\sum_{j=1}^{n} a_j > R$, т.е. все предметы в ранец положить невозможно. Необходимо положить в ранец набор предметов с максимальной суммарной ценностью. Введем n переменных: $x_j = 0$, если предмет с номером j не кладется в ранец, $x_j = 1$, если предмет с номером j кладется в ранец, $j = 1,2,...,n$. После введения этих переменных суммарная ценность и грузоподъемность соответственно имеют вид

$$f(x_1, x_2, ..., x_n) = \sum_{j=1}^{n} c_j x_j,$$

$$g(x_1, x_2, ..., x_n) = \sum_{j=1}^{n} a_j x_j.$$

Поэтому задача об одномерном ранце имеет вид

$$\sum_{j=1}^{n} c_j x_j \to \max, \tag{3.1}$$

$$\sum_{j-1}^{n} a_j x_j \le R, \qquad (3.2)$$

$$x_j \in \{0;1\},\ j = 1,2,...,n. \qquad (3.3)$$

Естественно предположить, что $c_j > 0, 0 < a_j \le R, j = 1,2,...,n$.

Множество допустимых решений этой задачи – это множество n -мерных булевых векторов $x = (x_1, x_2, ..., x_n)$, удовлетворяющих условию (3.2). Вместе с задачей (3.1)-(3.3) будем рассматривать задачу линейного программирования, в которой вместо условий (3.3) вводятся условия

$$0 \le x_j \le 1,\ j = 1,2,...,n. \qquad (3.4)$$

Существование допустимого решения. В задаче (3.1)-(3.3) допустимое решение всегда существует (например, нулевое). Если решить задачу линейного программирования (3.1)-(3.4) и в полученном оптимальном решении заменить дробные значения x_j нулевыми, то получим допустимое решение задачи (3.1)-(3.3).

Отсев подмножеств булевых векторов, не содержащих оптимальных решений. Рассмотрим вектор $\bar{x} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$, имеющий k единиц и $n-k$ нулей. Если вектор $\bar{x}$ является допустимым (т.е. условие (3.2.) выполнено), то можно исключить из дальнейшего рассмотрения 2^k векторов, получающихся из $\bar{x}$ заменой любого числа единиц нулями, так как для любого такого вектора x значение функции $f(x)$ будет меньше, чем $f(\bar{x})$. Если вектор $\bar{x}$ не является допустимым (т.е. условие (3.2) не выполняется), то можно исключить из дальнейшего рассмотрения 2^{n-k} векторов, которые получаются из $\bar{x}$ заменой любого числа нулей единицами, так как все эти векторы недопустимы.

Оценка точности приближенных булевых решений. Пусть $z^0 = (z_1^0, z_2^0, ..., z_n^0)$ - допустимое булево решение задачи (3.1)-(3.3); $x^0 = (x_1^0, x_2^0, ..., x_n^0)$ - оптимальное булево решение; $y^0 = (y_1^0, y_2^0, ..., y_n^0)$ - оптимальное решение задачи (3.1)-(3.4). Тогда имеет место следующее неравенство:

$$f\left(z^0\right) \le f\left(x^0\right) \le f\left(y^0\right). \tag{3.5}$$

Величина $\Delta = f\left(y^0\right) - f\left(z^0\right)$ есть гарантированная оценка отклонения приближенного решения z^0 от оптимума x^0.

Нахождение вектора y^0 связано с решением задачи линейного программирования. Далее будет показано, что задача (3.1)-(3.4) решается просто.

Алгоритм Данцига для линейной одномерной задачи о ранце.

Рассмотрим задачу линейного программирования (3.1)-(3.4) и опишем алгоритм нахождения оптимального решения $y^0 = \left(y_1^0, y_2^0, ..., y_n^0\right)$.

Теорема (правило Данцига).

Пусть переменные $x_j, 1 \le j \le n$ перенумерованы так, что $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_n$, где $\lambda_j = c_j / a_j$. Тогда оптимальное решение $y^0 = \left(y_1^0, y_2^0, ..., y_n^0\right)$ имеет вид

$$y_1^0 = y_2^0 = ... = y_{s-1}^0 = 1,$$

$$y_{s+1}^0 = y_{s+2}^0 = ... = y_n^0 = 0,$$

$$y_s^0 = \left(R - \sum_{j=1}^{s-1} a_j\right) / a_s,$$

где s определяется из условия

$$\sum_{j=1}^{s-1} a_j \le R < \sum_{j=1}^{s} a_j .$$

Смысл этого правила очевиден. Единичные значения следует последовательно назначать переменным, начиная с наибольшей удельной ценности (ценности на единицу веса), при этом

$$f\left(y^0\right) = \sum_{j=1}^{s-1} c_j + c_s \left(R - \sum_{j=1}^{s-1} a_j\right) / a_s .$$

Если все компоненты $y_j^0, j = 1,2,...,n$ целые (число дробных компонент не больше одной), то $y^0 = x^0$; если $0 < y_s^0 < 1$, то план z^0 получится при $y_s^0 = 0$, тогда

$$f\left(y^0\right)-f\left(z^0\right)=c_s\left(R-\sum_{j=1}^{s-1}a_j\right)/\,a_s\,.$$

Алгоритмы последовательного назначения единиц для приближенного решения задачи об одномерном ранце.

Прямые алгоритмы. В алгоритмах этого типа в качестве начального рассматривается нулевой вектор, в котором переменным присваиваются единичные значения в соответствии с заданными правилами.

Правило 1. При построении допустимого целочисленного решения z^0 следует назначать единичные значения в соответствии с последовательностью $\lambda_{j_1}\ge\lambda_{j_2}\ge\ldots\ge\lambda_{j_n}$, начиная с наибольшего значения. Если для некоторого λ_{j_k} такое назначение выполнить невозможно, то полагаем $z^0_{j_k}=0$ и переходим к номеру j_{k+1}. Процесс завершается после просмотра всех переменных или при нарушении условия(3.2). Это правило основано на теореме, приведенной выше.

Рассмотрим примеры.

Пример 3.1.

$6x_1+5x_2+2x_3+3x_4+4x_5+5x_6+4x_7+8x_8+7x_9+3x_{10}\to\max$,

$2x_1+2x_2+2x_3+x_4+2x_5+2x_6+2x_7+3x_8+3x_9+x_{10}\le 10$,

$x_j\in\{0,1\},\ j=1,2,\ldots,10$.

Р е ш е н и е. Вычислим значения λ_j:

$\lambda_1=3$, $\lambda_2=5/2$, $\lambda_3=1$, $\lambda_4=3$, $\lambda_5=2$, $\lambda_6=5/2$, $\lambda_7=2$, $\lambda_8=8/3$, $\lambda_9=7/3$, $\lambda_{10}=3$

. Дальнейшие вычисления сведены в табл. 3.1.

Таблица 3.1

λ_j	Значения переменных y_j^0	$f\left(y^0\right)$	$g\left(y^0\right)$
$\lambda_1=3$	$y_1^0=1$	6	2
$\lambda_4=3$	$y_4^0=1$	6+3=9	2+1=3
$\lambda_{10}=3$	$y_{10}^0=1$	9+3=12	3+1=4
$\lambda_8=8/3$	$y_8^0=1$	12+8=20	4+3=7

$\lambda_2 = 5/2$	$y_2^0 = 1$	20+5=25	7+2=9
$\lambda_6 = 5/2$	$y_6^0 = (10-9)/2 = 0{,}5$	25+0,5*5=27,5	9+0,5*2=10

Из табл. 3.1 получаем, что

$$y_3^0 = y_5^0 = y_7^0 = y_9^0 = 0\,.$$

Решение линейной задачи имеет вид

$$y^0 = (1;1;0;1;0;0{,}5;0;1;0;1)\,, \quad f\left(y^0\right) = 27{,}5\,.$$

Допустимое целочисленное решение имеет вид

$$z^0 = (1;1;0;1;0;0;0;1;0;1)\,, \quad f\left(z^0\right) = 25\,.$$

Поэтому можно записать $25 \le f\left(x^0\right) \le 27{,}5$.

Полученное отклонение $\Delta = 2{,}5$ приближенного решения от оптимума можно улучшить, если обратить внимание на то, что все значения c_j целые. Поэтому

$$25 \le f\left(x^0\right) \le 27\,.$$

Отметим, что в данном примере при $R = 9$ было бы получено точное решение x^0. Первое правило последовательного назначения единичных значений в рассмотренном примере дало достаточно хороший результат. Рассмотрим другой пример, в котором полученное по этому правилу решение z^0 сильно отличается от оптимального.

Пример 3.2.

$$5x_1 + 200x_2 \to \max\,,$$

$$5x_1 + 400x_2 \le 401\,,$$

$$x_j \in \{0;1\},\ j = 1,2\,.$$

Р е ш е н и е. Оптимальное решение: $x^0 = (0,1)$, $f\left(x^0\right) = 200$. Воспользуемся правилом Данцига:

$\lambda_1 = 1$, $\lambda_2 = 0{,}5$, $y^0 = (1;0{,}99)$, $f\left(y^0\right) = 203$, $z^0 = (1;0)$, $f\left(z^0\right) = 5$, $5 \le f\left(x^0\right) \le 203$.

Полученный «плохой» результат можно, по-видимому, объяснить большим разбросом в значениях элементов каждого из векторов $c = (c_1, c_2)$ и $a = (a_1, a_2)$. В первом примере такого разброса нет. Пример 3.2 указывает второе правило последовательного назначения единичных значений в следующей последовательности: $c_{j_1} \geq c_{j_2} \geq \ldots \geq c_{j_n}$. В примере 3.2 по второму правилу получаем

$$z^0 = (0;1),\ f(z^0) = 200,\ 200 \leq f(x^0) \leq 203.$$

Применим второе правило для примера 3.1. Вычисления сведены в табл. 3.2.

Таблица 3.2

c_j	Значения булевых переменных	$f(z^0)$	$g(z^0)$
$c_8 = 8$	$z_8^0 = 1$	8	3
$c_9 = 7$	$z_9^0 = 1$	15	6
$c_1 = 6$	$z_1^0 = 1$	21	8
$c_2 = 5$	$z_2^0 = 1$	26	10
	$z_3^0 = z_4^0 = z_5^0 = z_6^0 = z_7^0 = z_{10}^0 = 0$		

Получено допустимое целочисленное решение

$$z^0 = (1;1;0;0;0;0;0;1;1;0),$$

$$f(z^0) = 26;\quad 26 \leq f(x^0) \leq 27.$$

Сформулируем второе правило последовательного назначения единиц.

Правило 2. Назначения единичных значений выполняются в соответствии с последовательностью $c_{j_1} \geq c_{j_2} \geq \ldots \geq c_{j_n}$.

Можно предложить и другие правила. По-видимому, не представляется возможным указать наилучшее правило из заданного их набора для решения конкретной задачи. Но всегда возможно применение всех правил из набора к этой задаче и выбор лучшего из полученных результатов. Алгоритмы, основанные на описанных здесь правилах, получили название алгоритмов типа «greedy» (жадные, пожирающие алгоритмы). Основная особенность этих алгоритмов состоит в том, что глобальная оптимизация целевой функции заменяется многошаговой локальной оптимизацией на каждом шаге.

Рассмотрим модификации правил 1 и 2, которые в ряде случаев позволяют улучшить допустимые целочисленные решения z^0. Эти модификации предполагают фиксацию каждой из переменных последовательно на значениях 0 и 1, значения других переменных определяются с помощью правил 1 и 2.

В примере 3.1 положим $z_{10}^0 = 0$, остальные значения определим по правилу 1. Вычисления сведены в табл. 3.3.

Таблица 3.3

λ_j	Значения булевых переменных	$f(z^0)$	$g(z^0)$
$\lambda_{10} = 3$	$z_{10}^0 = 0$	0	0
$\lambda_1 = 3$	$z_1^0 = 1$	6	2
$\lambda_4 = 3$	$z_4^0 = 1$	9	3
$\lambda_8 = 8/3$	$z_8^0 = 1$	17	6
$\lambda_2 = 5/2$	$z_2^0 = 1$	22	8
$\lambda_6 = 5/2$	$z_6^0 = 1$	27	10
	$z_3^0 = z_5^0 = z_7^0 = z_9^0 = 0$		

Получено допустимое целочисленное решение

$z^0 = (1;1;0;1;0;1;0;1;0;0;)$, $f(z^0) = 27$,

которое в силу условия $27 \le f(x^0) \le 27$ оптимально, т.е. $x^0 = z^0$.

Рассмотрим применение правила 2 для этого же примера при $z_8^0 = 0$; здесь $c_8 = \max c_j$, $j = 1,2,...,10$.

Вычисления сведены в табл. 3.4.

Таблица3.4

c_j	*Значения булевых переменных*	$f(z^0)$	$g(z^0)$
$c_8 = 8$	$z_8^0 = 0$	*0*	*0*
$c_9 = 7$	$z_9^0 = 1$	*7*	*3*
$c_1 = 6$	$z_1^0 = 1$	*13*	*5*
$c_2 = 5$	$z_2^0 = 1$	*18*	*7*
$c_6 = 5$	$z_6^0 = 1$	*23*	*9*
$c_5 = 4$	$z_5^0 = 0$	*23*	*9*

$c_7 = 4$	$z_7^0 = 0$	*23*	*9*
$c_{10} = 1$	$z_{10}^0 = 1$	*26*	*10*
	$z_3^0 = z_4^0 = 0$		

Получено булево решение

$$z^0 = (1;1;0;0;0;1;0;0;1;1), \quad f(z^0) = 26,$$

совпадающее с полученным в табл. 3.2 по значению целевой функции.

Рассмотренные алгоритмы решения одномерной задачи о ранце получили название *эвристических.* Под эвристическими понимаются алгоритмы, основанные на правдоподобных, но не обоснованных строго предположениях о свойствах оптимального решения задачи. В первом правиле алгоритм точного решения задачи (3.1)-(3.4) переносится на формирование допустимого целочисленного решения задачи (3.1)-(3.3), в результате такого перенесения получается эвристический алгоритм для задачи (3.1)-(3.3). Полученное целочисленное решение может оказаться сильно отклоняющимся от оптимального. Причина такого поведения эвристического алгоритма состоит, по-видимому, в том, что любая эвристика верна лишь в некоторой, неизвестной, как правило, области изменения параметров задачи.

Двойственные алгоритмы. Рассмотренные ранее алгоритмы типа «greedy» часто называют прямыми. В алгоритмах этого типа формирование допустимого решения z^0 начинается с нулевого вектора, т.е. реализуется процедура подъема по допустимым целочисленным решениям. Формирование допустимого решения ω^0 в двойственных алгоритмах начинается с вектора $\bar{x}$, все компоненты которого равны 1. Замена единиц нулями начинается с наименьшего значения λ_j (или c_j) по возрастанию значений, т.е. реализуется процедура спуска по недопустимым целочисленным векторам.

В примере 3.1 положим $z_{10}^0 = 0$, остальные значения определим по правилу 1. Вычисления сведены в табл. 3.5.

Таблица 3.5

λ_j	Значения булевых переменных	$f(\bar{x})$	$g(\bar{x})$
	$\bar{x}_j = 1,\ j = 1,2,...,10$	47	20
$\lambda_3 = 1$	$\bar{x}_3 = 0$	45	18
$\lambda_5 = 2$	$\bar{x}_5 = 0$	41	16
$\lambda_7 = 2$	$\bar{x}_7 = 0$	37	14
$\lambda_9 = 7/3$	$\bar{x}_9 = 0$	30	11
$\lambda_6 = 5/2$	$\bar{x}_6 = 0$	25	9

Получено двойственное допустимое решение

$\omega^0 = (1;1;0;1;0;0;0;1;0;1)$, $f(z^0) = f(\omega^0) = 25$.

Из табл. 3.5 видно, что на последнем шаге можно получить $\bar{x}_2 = 0$, тогда двойственное решение имеет вид

$\omega^0 = (1;0;0;1;0;1;0;1;0;1)$, $f(\omega^0) = 25$, $\omega^0 \neq z^0$.

Для иллюстрации работы двойственного алгоритма рассмотрим пример 3.3.

Пример 3.3.

$6x_1 + 4x_2 + x_3 \to \max$,

$2x_1 + 2x_2 + x_3 \le 3$,

$x_j \in \{0;1\},\ j = 1,2,3.$

Р е ш е н и е.

Поскольку $\lambda_1 = 3, \lambda_2 = 2, \lambda_3 = 1$, то прямой алгоритм дает допустимое решение

$$z^0 = (1;0;1),\quad f(z^0) = 7,\quad y^0 = (1;0,5;0),\quad f(y^0) = 8.$$

Вычисления для двойственного алгоритма сведены в табл. 3.6.

Таблица 3.6

	Значения булевых переменных	$f(\bar{x})$	$g(\bar{x})$

λ_j	$\bar{x}_j = 1,\ j = 1,2,3$	11	5
$\lambda_3 = 1$	$\bar{x}_3 = 0$	10	4
$\lambda_2 = 2$	$\bar{x}_2 = 0$	6	2

Получено двойственное решение

$\omega^0 = (1;0;0)$, $f(\omega^0) = 6$, $\omega_j^0 \le z_j^0$, $f(\omega^0) < f(z^0)$.

Для изучения отношения $f(\omega^0)/f(z^0)$ рассмотрим пример 3.4.

Пример.3.4.

$2\gamma x_1 + 3x_2 + x_3 \to \max$,

$\gamma x_1 + 2x_2 + x_3 \le 2$,

$x_j \in \{0;1\}, 0 < \gamma \le 1,\ j = 1,2,3$.

Р е ш е н и е. Поскольку

$\lambda_1 = 2$, $\lambda_2 = 3/2$, $\lambda_3 = 1$,

то

$z^0 = (1;0;1)$, $f(z^0) = 2\gamma + 1$.

Для определения ω^0 вычисления сведены в табл.3.7.

Таблица 3.7

λ_j	Значения булевых переменных	$f(\bar{x})$	$g(\bar{x})$
	$\bar{x}_j = 1,\ j = 1,2,3$	$2\gamma + 4$	$3 + \gamma$
$\lambda_3 = 1$	$\bar{x}_3 = 0$	$2\gamma + 3$	$2 + \gamma$
$\lambda_2 = 3/2$	$\bar{x}_2 = 0$	2γ	γ

Получено решение для двойственного алгоритма

$\omega^0 = (1;0;0;)$, $f(\omega^0) = 2\gamma$, $\omega_j^0 \le z_j^0$, $j = 1,2,3;$

$$f(\omega^0) < f(z^0),\ \lim_{\gamma \to 0} \frac{f(\omega^0)}{f(z^0)} = \lim_{\gamma \to 0} \frac{2\gamma}{2\gamma + 1} = 0.$$

Наряду с задачей об одномерном ранце рассматривается задача о многомерном ранце [7].

Упражнения для самостоятельного решения

Методом Данцига решить задачи о ранце:

3.5. $17x_1+9x_2+11x_3+13x_4+15x_5+7x_6+2x_7 \to \max$,

$7x_1+3x_2+4x_3+5x_4+6x_5+2x_6+x_7 \le 15$,

$x_j \in \{0;1\},\ j=1,2,...,7$.

3.6. $60x_1+60x_2+40x_3+10x_4+20x_5+10x_6+3x_7 \to \max$,

$3x_1+5x_2+4x_3+x_4+4x_5+3x_6+x_7 \le 10$,

$x_j \in \{0;1\},\ j=1,2,...,7$.

4. Метод ветвей и границ

Суть метода ветвей и границ. Метод ветвей и границ относится к группе комбинаторных методов. Комбинаторные методы исходят из конечности числа допустимых планов задачи и заменяют полный перебор всех планов их частичным направленным перебором. Комбинаторные методы в значительно меньшей степени подвержены в процессе вычислений влиянию ошибок округления, поэтому являются более предпочтительными по сравнению с методами отсечения. Метод ветвей и границ – один из наиболее эффективных методов решения задач комбинаторного типа.

Рассмотрим общую задачу дискретного программирования

$$\max Z = f(x), \tag{4.1}$$

$$x \in \Omega, \tag{4.2}$$

где Ω - конечное множество допустимых планов.

1. Находим верхнюю границу (оценку) функции $f(x), x \in \Omega$, т.е. такое число $\varphi_0(\Omega)$, что для любых $x \in \Omega$

$$f(x) \le \varphi_0(\Omega).$$

Если при этом удается найти такой план x_0 задачи (4.1)-(4.2), для которого выполняется равенство

$$f\left(x_0\right) = \varphi_0\left(\Omega\right),$$

то x_0 - оптимальный план задачи (4.1)-(4.2).

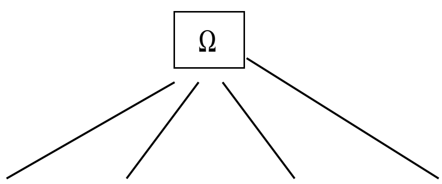
2. Если оптимальный план не найден, то некоторым способом разбиваем множество Ω на конечное число непересекающихся подмножеств $\Omega = \bigcup\limits_{r=1}^{r_1} \Omega_r^1$, $\bigcap\limits_{r=1}^{r_1} \Omega_r^1 = \varnothing$ и находим для каждого из этих подмножеств верхнюю границу $\varphi_1\left(\Omega_r^1\right)\left(r = 1,2,\ldots,r_1\right)$. Если при этом удастся найти такой план $x_l^1 \in \Omega_r^1\left(1 \le r \le r_1\right)$, что выполняется соотношение

$$f\left(x_l^1\right) = \varphi\left(\Omega_l^1\right) \ge \varphi\left(\Omega_r\right)\left(r = 1,2,\ldots,r_1\right),$$

то x_l^1- оптимальный план задачи (4.1)-(4.2.). Если же такой план не найден, то выбираем подмножество Ω_r^1 с наибольшей верхней границей (перспективное подмножество) и разбиваем его на несколько непересекающихся подмножеств $\Omega_s^2\left(s = 1,2,\ldots,s_1\right)$. Для каждого нового подмножества находим верхнюю границу $\varphi\left(\Omega_s^2\right)$. Если будет найден такой план x_k^2, что

$$f\left(x_k^2\right) = \varphi\left(\Omega_k^2\right) \ge \varphi\left(\Omega_s^2\right),$$

то x_k^2 - оптимальный план задачи. Если оптимальный план не найден, то дальнейшему ветвлению подвергаем подмножество с наибольшей верхней границей, и т.д. Процесс продолжается до получения оптимального плана. Способы ветвления и нахождения верхних границ выбираются для каждой конкретной задачи дискретного программирования. Процесс сопровождается построением дерева ветвления (рис.4.1).



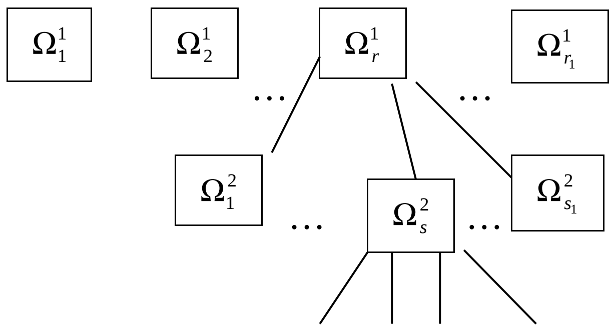


Рис. 4.1

Задача целочисленного (частично целочисленного) линейного программирования (2.1)-(2.4). В качестве верхней границы на множестве планов рассматривают значение целевой функции на оптимальном плане соответствующей ЗЛП (2.1)-(2.3). Пусть, далее, x_r - целочисленная переменная, значение x_r^* которой в оптимальном решении задачи (2.1)-(2.3) является дробным.

Интервал $\lfloor x_r^* \rfloor < x_r < \lfloor x_r^* \rfloor + 1$ не содержит допустимых целочисленных компонент решения. Поэтому допустимое целое значение x_r должно удовлетворять одному из неравенств: $x_r \le \lfloor x_r^* \rfloor$ или $x_r \ge \lfloor x_r^* \rfloor + 1$. Введение этих условий в задачу с отброшенным условием целочисленности порождает две не связанные между собой задачи. Говорят, что исходная задача разветвляется на две подзадачи. Затем каждая подзадача решается как ЗЛП с целевой функцией исходной задачи

$$\max Z = \sum_{j=1}^{n} c_j x_j; \qquad\qquad \max Z = \sum c_j x_j;$$

$$x \in \Omega, x_r \le \lfloor x_r^* \rfloor, , \qquad\qquad x \in \Omega, x_r \ge \lfloor x_r^* \rfloor + 1.$$

Если полученное оптимальное решение оказывается допустимым для целочисленной задачи, то его следует зафиксировать как наилучшее. При этом нет необходимости продолжать ветвление подзадачи, поскольку улучшить полученное решение не удастся. В противном случае подзадача должна быть разбита на две подзадачи и т.д. Как только полученное допустимое целочисленное решение одной из подзадач оказывается лучше имеющегося, оно фиксируется вместо зафиксированного ранее. Процесс ветвления продолжается до тех пор, пока каждая подзадача не приведет к целочисленному решению или пока не будет установлена невозможность улучшения имеющегося решения.

Пример 4.1.

$\max Z = 2x_1 + 3x_2$;

$5x_1 + 7x_2 \le 35,$
$4x_1 + 9x_2 \le 36,$

$$x_1 \ge 0, x_2 \ge 0, x_1, x_2 - \textit{целые.}$$

Р е ш е н и е. Решим исходную задачу (задачу 0) как ЗЛП с отброшенным условием целочисленности. Оптимальное решение достигается в точке $x^*_{нц} = \left(3\frac{12}{17}; 2\frac{6}{17}\right)$.

Так как обе переменные принимают нецелые значения, то любая из них может быть выбрана в качестве переменной, продолжающей процесс ветвления.

Выбор, например, переменной x_2 порождает две подзадачи, связанные с условием $x_2 \le \lfloor x_2^* \rfloor$ или $x_2 \ge \lfloor x_2^* \rfloor + 1$. Так как $\lfloor x_2^* \rfloor = \lfloor 40/17 \rfloor = 2$, имеем две подзадачи 1.1 и 1.2:

a. 1.2

$\max Z = 2x_1 + 3x_2$;	$\max Z = 2x_1 + 3x_2$;
$5x_1 + 7x_2 \le 35,$	$5x_1 + 7x_2 \le 35,$
$4x_1 + 9x_2 \le 36,$	$4x_1 + 9x_2 \le 36,$
$x_2 \le 2,$	$x_2 \ge 3,$
$x_1, x_2 \ge 0 - \textit{целые},$	$x_1, x_2 \ge 0 - \textit{целые.}$

Порожденные подзадачи содержат все допустимые целочисленные решения исходной задачи, т.е. исходное множество допустимых целочисленных решений остается неизменным в процессе ветвления. Их оптимальные решения:

$$x^*_{нц} = \left(4\frac{1}{5};2\right), Z^* = 14\frac{2}{5}; \qquad x^*_{нц} = \left(2\frac{1}{4};3\right), Z^* = 13\frac{1}{2}$$ соответственно.

На следующем шаге осуществляется выбор одной из подзадач 1.1 или 1.2 для рассмотрения и при необходимости для дальнейшего ветвления. Отметим, что не существует точных способов реализации указанного выбора. Причем выбор различных альтернатив приводит к разным последовательностям подзадач, а, следовательно, к различным количествам итераций, обеспечивающих получение оптимального целочисленного решения.

Предположим, что в первую очередь рассматривается задача 1.1. Оптимальное решение этой задачи достигается в точке $x^*_{нц} = \left(4\frac{1}{2};2\right)$. Так как значение $x_1^* = 4\frac{1}{2}$ остается нецелым, задача 1.1 порождает подзадачи 2.1 и 2.2 с дополнительными ограничениями соответственно $x_1 \le \left[x_1^*\right]$, т.е. $x_1 \le 4$ и $x_1 \ge \left[x_1^*\right]+1$, т.е. $x_1 \ge 5$:

2.1	2.2
$\max Z = 2x_1 + 3x_2;$	$\max Z = 2x_1 + 3x_2;$
$5x_1 + 7x_2 \le 35,$	$5x_1 + 7x_2 \le 35,$
$4x_1 + 9x_2 \le 36,$	$4x_1 + 9x_2 \le 36,$
$x_1 \le 4, x_2 \le 2,$	$x_1 \ge 5, x_2 \le 2,$
$x_1, x_2 \ge 0 - целые,$	$x_1, x_2 \ge 0 - целые$.

Оптимальные решения этих подзадач:

$$x^*_{нц} = (4;2), Z^* = 14 \quad и \quad x^*_{нц} = \left(5;1\frac{3}{7}\right), Z^* = 14\frac{2}{7}$$ соответственно.

Наличие у подзадачи 2.1 целочисленного решения не означает, что найдено оптимальное целочисленное решение исходной задачи 0, потому что еще не решены подзадачи 1.2 и 2.2, которые могут дать лучшее решение, чем 2.1. Целочисленное решение подзадачи 2.1 определяет нижнюю границу $Z = 14$ значений целевой функции. Нет необходимости рассматривать те последующие подзадачи, для которых оптимальные значения Z меньше указанной нижней границы.

Обратимся к подзадаче 1.2. Для нее $Z^* = 13{,}5$, что не превышает значения $Z = 14$, поэтому поиск вдоль ветви $x_2 \geq 3$ следует прекратить.

Исследуем вершину 2.2, которой соответствует $Z^* = 14\frac{2}{7}$. Несмотря на то, что полученное значение превышает нижнюю границу $Z = 14$, дальнейшее ветвление осуществлять нецелесообразно, поскольку $14\frac{2}{7} - 14 < 1$ и все коэффициенты целевой функции целые. Итак, найдено оптимальное решение задачи: $x_ц^* = (4;2), Z(x_ц^*) = 14$. К нему привела цепочка задач $0 \to 1.1 \to 1.2 \to 2.1 \to 2.2$.

Если для исследования выбрать подзадачу 1.2, то получим следующие оптимальные целочисленные решения: $x_ц^* = (2;3), Z^* = 13$ и $x_ц^* = (0;4), Z^* = 12$. Они явно хуже, чем решение подзадачи 2.1.

Упражнения для самостоятельного решения

Методом ветвей и границ найти оптимальные решения задач:

4.2. $2x_1 + 3x_2 \to \max$;

$2x_1 + 2x_2 \geq 1$,

$4x_1 - x_2 \leq 15$,

$x_1 + 3x_2 \leq 16$,

$x_1, x_2 \geq 0 - \textit{целые}$.

4.3. $3x_1 + x_2 \to \min$;

$-4x_1 + x_2 \leq 29$,

$3x_1 - x_2 \leq 15$,

$5x_1 + 2x_2 \geq 38$,

$x_1, x_2 \geq 0 - \textit{целые}$.

Список использованной литературы

1. Акулич И.Л. Математическое программирование в примерах и задачах. – М.: Высш.шк., 1993.

2. Кузнецов А.В., Холод Н.И. Математическое программирование. – Минск: Высш.шк., 1984.

3. Кузнецов А.В., Холод Н.И., Костевич Л.С. Руководство к решению задач по математическому программированию. – Минск: Высш.шк., 2001.

4. Морозов В.В. и др. Исследование операций в задачах и упражнениях. – М.: Высш.шк., 1986.

5. Саати Т.Л. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. – М., Мир, 1973.

6. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации. – Киев: Наукова думка, 1988.

7. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование. –М.: ФИЗМАТЛИТ, 2002.

8. Схрейвер А. Теория линейного и целочисленного программирования. Т.2. – М.: Мир, 1991.

Подписано в печать 14.10.2009.

Усл. печ. л. 1,75. Тираж 10 экз. Заказ №443.

Тверской государственный университет