



Методы оптимизации

**9. ЦЕЛЕВОЕ
ПРОГРАММИРОВАНИЕ**

9.1. НЕСКОЛЬКО ЦЕЛЕВЫХ ФУНКЦИЙ

Модели линейного программирования, рассмотренные в предыдущих главах, предполагали оптимизацию только *одной целевой функции*. Но возможны ситуации, когда в модели присутствует несколько (возможно, конфликтующих между собой) целевых функций. Например, честолюбивый политик может пообещать уменьшение национального долга и, одновременно, снижение ставок налогов. В таких ситуациях иногда невозможно найти единственное решение, оптимизирующее все конфликтующие целевые функции. Поэтому нужно искать *компромиссное* решение, учитывающее "важность" каждой целевой функции.

В этой главе представлены методы целевого программирования (многокритериальной оптимизации) для решения задач линейного программирования с несколькими целевыми функциями. Задачи целевого программирования являются только подклассом задач, решаемых методами многокритериальной оптимизации. Подходы и методы, применяемые в многокритериальной оптимизации, отнюдь не исчерпываются теми методами, которые описаны в этой главе. Основное назначение этих методов — преобразование исходной задачи с несколькими целевыми функциями в задачу ЛП с одной целевой функцией. После решения преобразованной задачи получаем так **называемое эффективное решение**, поскольку может не существовать оптимального решения, доставляющего оптимум *всем* частным целевым функциям исходной задачи.



9.2. ФОРМУЛИРОВКА ЗАДАЧИ ЦЕЛЕВОГО ПРОГРАММИРОВАНИЯ

Рассмотрим пример, приводящий к задаче целевого программирования.

Пример 7.2-1

Файрвилл — небольшой городок, в котором проживает около 20 тысяч жителей. Предположим, городской совет разрабатывает ставки местного налогообложения. Ежегодная база налогообложения недвижимости составляет 550 миллионов долларов. Ежегодная база налогообложения розничных и оптовых продаж составляет 35 и 55 миллионов долларов соответственно. Ежегодное потребление городом бензина оценивается в 7.5 миллионов галлонов. Городской совет планирует разработать систему налоговых ставок, основанную на перечисленных базах налогообложения и учитывающую следующие ограничения и требования.

1. Налоговые Поступления должны составить не менее 16 миллионов долларов от всех баз налогообложения.
2. Налог с розничных продаж не может превышать 10% от суммы всех собираемых налогов.
3. Налог с оптовых продаж не может превышать 20% от суммы всех налогов.
4. Налог на бензин не может превышать 2 центов за галлон.



Обозначим через x_n , x_p и x_o ставки налогов (выраженные десятичными дробями) на недвижимость, розничную и оптовую торговлю соответственно, а через x_b — налог на бензин, выраженный в центах на галлон. Пожелания городского совета можно записать следующим образом.

$$550x_n + 35x_p + 55x_o + 0,075x_b \geq 16 \text{ (Суммарные налоговые поступления)}$$

$$35x_p \leq 0,1(550x_n + 35x_p + 55x_o + 0,075x_b) \text{ (Налог на розничную торговлю)}$$

$$55x_o \leq 0,2(550x_n + 35x_p + 55x_o + 0,075x_b) \text{ (Налог на оптовую торговлю)}$$

$$x_b \leq 2, \text{ (Налог на бензин)}$$

$$x_n, x_p, x_o, x_b \geq 0.$$

После упрощения получаем три ограничения.

$$550x_n + 35x_p + 55x_o + 0,075x_b \geq 16,$$

$$55x_n - 31,5x_p + 5,5x_o + 0,0075x_b \geq 0,$$

$$110x_n + 7x_p - 44x_o + 0,015x_b \geq 0,$$

$$x_b \leq 2,$$

$$x_n, x_p, x_o, x_b \geq 0.$$

Каждое из этих неравенств представляет одну из целей городского совета, которую желательно добиться. Но эти цели могут конфликтовать друг с другом, и в лучшем случае мы можем попытаться достичь какого-нибудь компромиссного решения.

Способ, которым в целевом программировании достигается компромиссное решение, заключается в следующем. Сначала каждое неравенство преобразуется в более гибкую частную задачу, в рамках которой можно удовлетворить данное ограничение. В нашем случае эти частные задачи записываются так:

$$550x_n + 35x_p + 55x_o + 0,075x_b + s_1^+ - s_1^- = 16,$$

$$55x_n - 31,5x_p + 5,5x_o + 0,0075x_b + s_2^+ - s_2^- = 0,$$

$$110x_n + 7x_p - 44x_o + 0,015x_b + s_3^+ - s_3^- = 0,$$

$$x_b + s_4^+ - s_4^- = 2,$$

$$x_n, x_p, x_o, x_b \geq 0.$$

$$s_i^+, s_i^- \geq 0, i = 1, 2, 3, 4.$$



Каждое из этих неравенств представляет одну из целей городского совета, которую желательно добиться. Но эти цели могут конфликтовать друг с другом, и в лучшем случае мы можем попытаться достичь какого-нибудь компромиссного решения.

Способ, которым в целевом программировании достигается компромиссное решение, заключается в следующем. Сначала каждое неравенство преобразуется в более гибкую частную задачу, в рамках которой можно удовлетворить данное ограничение. В нашем случае эти частные задачи записываются так:

$$\begin{aligned} 550x_n + 35x_p + 55x_o + 0,075x_b + s_1^+ - s_1^- &= 16, \\ 55x_n - 31,5x_p + 5,5x_o + 0,0075x_b + s_2^+ - s_2^- &= 0, \\ 110x_n + 7x_p - 44x_o + 0,015x_b + s_3^+ - s_3^- &= 0, \\ x_b + s_4^+ - s_4^- &= 2, \\ x_n, x_p, x_o, x_b &\geq 0. \\ s_i^+, s_i^- &\geq 0, i = 1, 2, 3, 4. \end{aligned}$$

Неотрицательные переменные s_i^+ и s_i^- называются **отклоняющимися**, поскольку они показывают отклонение значений левых частей ограничений от соответствующих величин правых частей этих же ограничений.

Отклоняющие переменные s_i^+ и s_i^- зависимы по определению, поэтому они обе одновременно не могут быть базисными. Это означает, что, на любом этапе решения задачи одним из симплексных методов, только *одна* из пары отклоняющих переменных может принимать положительное значение. Если исходное i -е ограничение является неравенством типа " \leq " и $s_i^+ > 0$, то это ограничение выполняется. Если же $s_i^- > 0$, то данное ограничение не выполняется. Таким образом, определенные значения отклоняющих переменных s_i^+ и s_i^- либо удовлетворяют i -е ограничение, либо нет. Это та гибкость, которая позволяет целевому программированию достичь компромиссного решения. Естественно, хорошее компромиссное решение минимизирует число невыполняемых ограничений.

В нашем примере первые три ограничения являются неравенствами типа " \geq ", а четвертое — неравенством типа " \leq ". Вследствие этого положительные значения отклоняющих переменных s_1^+, s_2^+, s_3^+ и s_4^- будут указывать на то, что соответствующие ограничения не выполняются. Поэтому ведется поиск такого компромиссного решения, которое будет удовлетворять по возможности большему числу следующих частных целей (целевых функций).

Минимизировать $G_1 = s_1^+$

Минимизировать $G_2 = s_2^+$

Минимизировать $G_3 = s_3^+$

Минимизировать $G_4 = s_4^-$

Как оптимизировать модель, имеющую несколько конфликтующих целевых функций? Для этого разработано множество разнообразных методов. В данной главе рассмотрим (1) метод весовых коэффициентов и (2) метод приоритетов. Оба метода основаны на приведении множества частных целевых функций к одной целевой функции



9.3. АЛГОРИТМЫ ЦЕЛЕВРГО ПРОГРАММИРОВАНИЯ

В этом разделе представлены два метода решения задач целевого программирования. Оба метода основаны на сведении множества частных целей к одной целевой функции. В методе весовых **коэффициентов** единственная целевая функция формируется как взвешенная сумма исходных частных целевых функций. В методе приоритетов на частные цели устанавливаются приоритеты в порядке их важности. Исходная задача решается путем последовательного решения ряда задач ЛП с одной целевой функцией таким образом, что решение задачи с низкоприоритетной целью не может "испортить" оптимального значения целевой функции с более высоким приоритетом.

Эти методы различны по своей природе и в общем случае дают оптимальные решения, не совпадающие между собой. Вместе с тем нельзя сказать, что один из этих методов лучше другого; в сущности, они предназначены для решения задач с разными предпочтениями в процессе принятия решений.

Метод весовых коэффициентов

Предположим, что модель целевого программирования имеет n целей, каждая из которых имеет следующий вид.

Минимизировать $G_i, i = 1, 2, \dots, n$.

В методе весовых коэффициентов обобщенная целевая функция определяется следующим образом.

Минимизировать $z = w_1 G_1 + w_2 G_2 + \dots + w_n G_n$

Здесь $w_i (i = 1, 2, \dots, n)$ — положительные весовые коэффициенты, которые отображают предпочтения, отдаваемые каждой цели. Например, случай $w_1 = 1$ для всех i говорит о равнозначности всех целей. Задание значений весовым коэффициентам очень субъективно. В настоящее время разработаны различные методы, которые уменьшают субъективный фактор при определении весовых коэффициентов.



Пример 7.3-1

Новое рекламное агентство, в составе которого 10 рекламных агентов, получило контракт на рекламу нового продукта. Агентство может провести рекламную акцию на радио и телевидении. В следующей таблице приведены данные о количестве людей, охватываемых тем или иным видом рекламы, стоимость этой рекламы и количество необходимых рекламных агентов. Все эти данные отнесены к одной минуте рекламного времени.

	Радио	Телевидение
Рекламная аудитория (млн. чел)	4	8
Стоимость (тыс. доллар)	8	24
Количество рекламных агентов	1	2

Реклама на радио и телевидении должна охватить не менее 45 миллионов человек (так называемая рекламная аудитория), но контракт запрещает использовать более 6 минут рекламы на радио. Рекламное агентство может выделить на этот проект бюджет, не превышающий \$100 000. Сколько минут рекламного времени агентство должно купить на радио и сколько на телевидении?

Обозначим через x_1 и x_2 количество минут рекламного времени, закупленного соответственно на радио и телевидении. Для данной задачи целевого программирования можно задать следующие частные целевые функции.

Минимизировать $G_1 = s_1^+$ (для выполнения условия по рекламной аудитории),

Минимизировать $G_2 = s_1^-$ (для выполнения условия по бюджету)

при выполнении ограничений

$$4x_1 + 8x_2 + s_1^+ - s_1^- = 45 \text{ (условие по рекламной аудитории),}$$

$$8x_1 + 24x_2 + s_2^+ - s_2^- = 100 \text{ (условия по бюджету),}$$

$$x_1 + 2x_2 \leq 10 \text{ (ограничение по рекламным агентам),}$$

$$x_1 \leq 6 \text{ (ограничение на рекламу по радио),}$$

$$x_1, x_2, s_1^+, s_1^-, s_2^+, s_2^- \geq 0.$$



Менеджеры рекламного агентства считают, что выполнение условия по объему рекламной аудитории в два раза важнее, чем выполнение условия по бюджету. Поэтому обобщенная целевая функция будет записана следующим образом.

Минимизировать $z = 2G_1 + G_2 = 2s_1^+ + s_2^-$.

Оптимальное решение этой задачи, следующее: $z = 10$, $x_1 = 5$ минут, $x_2 = 2.5$ минуты, $s_1^+ = 5$ миллионов человек. Остальные переменные равны нулю.

Тот факт, что оптимальное значение целевой функции не равно нулю, указывает, что, по крайней мере, одна из исходных целевых функций не достигла своего оптимального значения. Действительно, так как $s_1^+ = 5$, значит, объем рекламной аудитории меньше запланированного на 5 миллионов. При этом условие по бюджету выполнено, поскольку $s_2^- = 0$,

Еще раз повторим, что методы целевого программирования позволяют получить только *эффективное решение* задачи, которое не всегда будет оптимальным.

Например, решение $x_1 = 6$ и $x_2 = 2$ дает такой же объем рекламной аудитории ($4 \times 6 + 8 \times 2 = 40$ миллионов человек), но при меньшей стоимости рекламной кампании ($8 \times 6 + 24 \times 2 = \$96\,000$). Существенно, что методы целевого программирования в общем случае не находят оптимум каждой целевой функции исходной модели. Этот "дефект" методов целевого программирования поднимает общий вопрос о "жизнеспособности" целевого программирования в качестве технологии оптимизации.

Метод приоритетов

В методе приоритетов n частных целевых функций ранжируются в порядке их важности, так как их оценивает специалист по принятию решений, т.е.

Минимизировать $G_1 = p_1$ (наивысший приоритет),

...

Минимизировать $G_n = p_n$ (наинизший приоритет).

Переменные p_i — это компоненты отклоняющихся переменных, т.е. s_i^+ или s_i^- , которые определяют i -ю целевую функцию. Например, в задаче из примера 7.3-1 $p_1 = s_1^+$ и $p_2 = s_2^-$.



В методе приоритетов поочередно решаются задачи с одной целевой функцией, начиная с задачи с целевой функцией, имеющей наивысший приоритет, и заканчивая задачей с целевой функцией, имеющей наименьший приоритет. *В процессе решения последовательных задач решение задачи с целевой функцией, имеющей более низкий приоритет, не может ухудшить полученные ранее решения задач с целевой функцией, имеющих более высокий приоритет.* Это означает, что если $z(G_i)$ — оптимальное значение целевой функции G_i , то для всех $i \geq 1$ оптимизация любой целевой функции G_j ($j > i$) с меньшим приоритетом не может ухудшить значение $z(G_i)$.

В *литературе по целевому* программированию описан "специальный" симплекс-метод, который гарантирует неухудшаемость решений задач с целевыми функциями более высокого приоритета. Этот метод использует **правило** исключения **столбцов**, которое применяется для удаления из оптимальной симплекс-таблицы задачи с целевой функцией G_k *небазисной* переменной x_j , с $z_j - c_j \neq 0$ до начала решения задачи с целевой функцией G_{k+1} . Это правило распознает, что небазисная переменная x_j , если она получит ненулевое значение, может ухудшить (но никогда не улучшит) оптимальное значение задачи с целевой функцией, имеющей более высокий приоритет.

К сожалению, этот метод изменения симплекс-таблиц требует более сложных вычислений, чем необходимо на самом деле. Мы покажем, что такого же результата можно достигнуть более простым способом.

Шаг 0. Определяем частные целевые функции задачи и ранжируем их в порядке приоритетов: $G_1 = p_1 > G_2 = p_2 > \dots > G_n = p_n$. Положим $i = 1$.

Шаг i. Решаем i -ю задачу ЛП с целевой функцией G_i . Обозначим через p_i^* полученное оптимальное значение отклоняющей переменной p_i . Если $i = n$, вычисления заканчиваются, поскольку решена последняя n -я задача. В противном случае вводим в задачу новое ограничение $p_i = p_i^*$, тогда значение p_i не сможет измениться при решении последующих задач. Полагаем $i = i + 1$ и повторяем шаг i .

Последовательное введение дополнительных ограничений вида $p_i = p_i^*$ теоретически не так "элегантно", как *правило исключения столбцов*. Однако это приводит точно к такому же результату. Кроме того, обоснование введения дополнительных ограничений совершенно очевидно и понятно.



Определенным аргументом в пользу правила исключения столбцов может служить то, что при использовании этого правила происходит удаление переменной, т.е. уменьшается размерность задачи. В то же время описанная выше процедура увеличивает размерность задачи при добавлении новых ограничений. Но если повнимательнее присмотреться к этим ограничениям (вида $p_i = p_i^*$), то легко изменить вычисления таким образом, чтобы это ограничение учитывалось непосредственно путем подстановки значения p_i^* вместо переменной p_i , что также уменьшает количество переменных в задаче. (Можно также использовать метод решения задач ЛП с ограниченными переменными для выполнения замены $p_i = p_i^*$, предполагая при этом ограничение вида $p_i \leq p_i^*$) С этой точки зрения правило исключения столбцов, если отвлечься от его теоретической привлекательности, не имеет особых вычислительных преимуществ по сравнению с описанной выше процедурой. В следующем примере покажем использование описанной выше упрощенной процедуры решения задач с приоритетами.

Пример 7.3-2

Решим методом приоритетов задачу из примера 7.3-1. Предположим, что наибольший приоритет имеет частная целевая функция, соответствующая условию, накладываемому на объем рекламной аудитории.

Шаг 0. $G_1 > G_2$, где

G_1 : Минимизировать s_1^+ (условие по рекламной аудитории),

G_2 : Минимизировать s_2^- (условие по бюджету).

Шаг 1. Решаем первую задачу ЛП.

Минимизировать $G_1 = s_1^+$

при выполнении ограничений

$$4x_1 + 8x_2 + s_1^+ - s_1^- = 45 \text{ (условие по рекламной аудитории),}$$

$$8x_1 + 24x_2 + s_2^+ - s_2^- = 100 \text{ (условие по бюджету),}$$

$$x_1 + 2x_2 \leq 10 \text{ (ограничение по рекламным агентам),}$$

$$x_1 \leq 6 \text{ (ограничение на рекламу по радио),}$$

$$x_1, x_2, s_1^+, s_1^-, s_2^+, s_2^- \geq 0.$$

Оптимальное решение этой задачи составляет $x_1 = 5$ минут, $x_2 = 2.5$ минуты, $s_1^+ = 5$ миллионов человек, остальные переменные равны нулю. Решение показывает, что условие по объему рекламной аудитории не выполняется с дефицитом в 5 млн. человек. В этой задаче мы имеем $p_1 = s_1^+$. Поэтому в следующей задаче добавим ограничение $s_1^+ = 5$.

Шаг 2. Теперь необходимо решить вторую задачу ЛП.

Минимизировать $G_2 = s_2^-$

при выполнении тех же ограничений, что и в предыдущей задаче, плюс дополнительное ограничение $s_1^+ = 5$.

В данном случае в решении второй задачи нет необходимости, поскольку уже в решении первой имеем $s_2^- = 0$. Следовательно, решение первой задачи автоматически является оптимальным решением второй. Решение $s_2^- = 0$ показывает, что ограничение, касающееся бюджета рекламной компании, выполняется.

Дополнительное ограничение $s_1^+ = 5$ можно также учесть путем подстановки значения 5 вместо переменной s_1^+ в первое ограничение. В результате правая часть этого неравенства изменится со значения 45 на 40. Получим следующую задачу ЛП.

Минимизировать $G_2 = s_2^-$

при ограничениях

$$\begin{aligned}4x_1 + 8x_2 - s_1^- &= 40, \\8x_1 + 24x_2 + s_2^+ - s_2^- &= 100, \\x_1 + 2x_2 &\leq 10, \\x_1 &\leq 6, \\x_1, x_2, s_1^+, s_1^-, s_2^+, s_2^- &\geq 0.\end{aligned}$$

В новой формулировке этой задачи на одну переменную меньше, чем в первой задаче.

Теперь мы используем ту же задачу, чтобы показать, что наилучшее решение получается тогда, когда в методе приоритетов используется *оптимизация* "настоящих" целевых функций, а не тех целевых функций, которые строятся только для того, чтобы выполнялись определенные ограничения. Следующий пример также демонстрирует правило исключения столбцов при решении задач целевого программирования.



Пример 7.3-3

Цели, поставленные в задаче из примера 7.3-1, можно переформулировать следующим образом.

Цель 1. Максимизировать объем рекламной аудитории (P_1).

Цель 2. Минимизировать стоимость рекламной кампании (P_2).

Математически эти цели можно выразить с помощью следующих целевых функций.

Максимизировать $P_1 = 4x_1 + 8x_2$,

Минимизировать $P_2 = 8x_1 + 24x_2$.

Отдельные ограничения на желаемый объем рекламной аудитории и стоимость рекламной кампании, которые использовались в примерах 7.3-1 и 7.3-2, в данном случае излишни, поскольку для этих величин мы получим границы после решения соответствующих задач.

Получили новую задачу.

Максимизировать $P_1 = 4x_1 + 8x_2$,

Минимизировать $P_2 = 8x_1 + 24x_2$

при ограничениях

$$x_1 + 2x_2 \leq 10,$$

$$x_1 \leq 6,$$

$$x_1, x_2 \geq 0.$$

Сначала решим эту задачу с помощью процедуры, описанной в примере 7.3-2,

Шаг 1. Решаем первую задачу ЛП.

Максимизировать $P_1 = 4x_1 + 8x_2$,

при ограничениях

$$x_1 + 2x_2 \leq 10,$$

$$x_1 \leq 6,$$

$$x_1, x_2 \geq 0.$$

Оптимальное решение этой задачи составляет $x_1 = 0$, $x_2 = 5$ и $P_1 = 40$. Отсюда видно, что объем рекламной аудитории не может превысить 40 миллионов человек.



Шаг 2. Добавим ограничение $4x_1 + 8x_2 \geq 40$ которое гарантирует, что решение, полученное на предыдущем шаге, не будет ухудшено, и решаем следующую задачу ЛП.

$$\text{Минимизировать } P_2 = 8x_1 + 24x_2$$

при ограничениях

$$x_1 + 2x_2 \leq 10,$$

$$x_1 \leq 6,$$

$$4x_1 + 8x_2 \geq 40,$$

$$x_1, x_2 \geq 0.$$

Оптимальное решение этой задачи: $P_2 = 96\ 000$, $x_1 = 6$ минут и $x_2 = 2$ минуты. Мы получили тот же объем рекламной аудитории ($P_1 = 40$ млн. чел.), но за меньшую стоимость. Это результат того, что здесь мы искали оптимальные значения соответствующих величин, а не просто удовлетворяли ограничениям, как в примере 7.3-2.

Теперь решим ту же задачу, используя правило исключения столбцов. Это правило применяется последовательно к строкам симплекс-таблицы, соответствующим частным целевым функциям.

Первая задача ЛП (Максимизация объема рекламной аудитории). При решении этой задачи симплекс-таблица содержит строки, соответствующие как целевой функции P_1 , так и целевой функции P_2 . Строка целевой функции P_2 пока играет пассивную роль, но будет изменена перед решением второй задачи ЛП.

Первая задача решается за две симплексные итерации, как показано в следующей таблице.

Базис	x_1	x_2	s_1	s_2	Решение
P_1	-4	-8	0	0	0
P_2	-8	-24	0	0	0
s_1	1	2	1	0	10
s_2	1	0	0	1	6
P_1	0	0	4	0	40
P_2	4	0	12	0	120
x_2	1/2	1	1/2	0	5
s_2	1	0	0	1	6



Нижняя часть этой таблицы показывает оптимальное решение $x_1 = 0$, $x_2 = 5$ и $P_1 = 40$ первой задачи.

Правило исключения столбцов применяется перед решением второй задачи для удаления из симплекс-таблицы с оптимальным решением первой задачи *небазисной* переменной x_j для которой $z_j - c_j \neq 0$. Такие переменные, приняв положительные значения в задаче с более низким приоритетом, ухудшают решение задач с более высоким приоритетом.

Вторая задача ЛП (Минимизация стоимости рекламной кампании). Правило исключения столбцов удаляет переменную s_1 , для которой $z_1 - c_2 = 4$. Из P_2 - строки приведенной выше симплекс-таблицы видно, что если не удалить переменную s_1 , то на первой итерации решения второй задачи она должна войти в базис, при этом из базиса будет исключена переменная x_2 . После этого будет получено оптимальное решение второй задачи (в этом решении $x_1 = x_2 = 0$), которое ухудшает оптимальное решение первой, поскольку теперь $P_1 = 0$ вместо $P_1 = 40$, как было ранее.

В данном случае вторая задача ЛП является задачей минимизации. После удаления переменной s_1 в базис вводится небазисная переменная x_1 , со значением разности $z_j - c_j$ равным 4 (> 0), что может улучшить значение целевой функции P_2 . В следующей симплекс-таблице показаны две итерации решения второй задачи ЛП. P_1 - строку можно удалить из этой таблицы, так как она не участвует в процессе нахождения оптимального решения задачи с целевой функцией P_2 .

Итерация	Базис	x_1	x_2	s_1	s_2	Решение
1	P_1					40
	P_2	4	0		0	120
	x_2	1/2	1		0	5
	s_2	1	0		1	6
2	P_1					40
	P_2	0	0		-4	96
	x_2	0	1		-1/2	2
	x_1	1	0		1	6

Полученное здесь оптимальное решение ($x_1 = 6$, $x_2 = 2$) со значениями целевых функций $P_1 = 40$ и $P_2 = 96$ такое же, как и полученное ранее.

7.4. ЗАКЛЮЧЕНИЕ

Задачи целевого программирования — это задачи принятия решений, имеющих несколько, возможно противоречивых, целевых функций. Основное назначение методов целевого программирования заключается в сведении задачи с несколькими целевыми функциями к одной или нескольким задачам ЛП с одной целевой функцией. Качество конечного решения задачи целевого программирования определяется способом ранжирования частных целевых функций по степени их важности, а также степенью гибкости множества ограничений задачи. Поэтому методы целевого программирования дают *эффективное решение*, которое только пытается учесть все частные целевые функции, но, как правило, не совпадает с их оптимальными решениями.

