

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет информационных технологий

# Е. В. Алексеева

**ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ЦЕЛОЧИСЛЕННОГО**

**ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ. ПРИМЕРЫ И ЗАДАЧИ**

Учебное пособие

Новосибирск 2012

УДК 519.8(075.8) ББК В183я73-1 A 471

Алексеева Е. В. Построение математических моделей целочисленного линейно- го программирования. Примеры и задачи: Учеб. пособие / Новосиб. гос. ун-т. Новосибирск, 2012. 131 с.

ISBN

Пособие предназначено для студентов и магистрантов Новосибирского го- сударственного университета, изучающих дисциплины «Теория принятия ре- шений» и «Исследование операций». Материал, содержащийся в пособии, яв- ляется частью основных лекционных курсов и семинарских занятий по этим дисциплинам.

Рецензент

д-р физ.-мат. наук, проф. Ю. А. Кочетов

ISBN

⃝c Новосибирский государственный

университет, 2012

⃝c Алексеева Е. В., 2012

**Оглавление**

1. [Введение 4](#_TOC_250030)
2. [Моделирование с помощью булевых переменных 12](#_TOC_250029)
   1. [Примеры математических моделей 13](#_TOC_250028)
   2. [Правила моделирования логических импликаций 15](#_TOC_250027)
   3. [Моделирование свойств логических отношений 22](#_TOC_250026)
   4. [Моделирование выбора минимального элемента 24](#_TOC_250025)
   5. [Моделирование взаимоисключающих событий 25](#_TOC_250024)
   6. [Линеаризация в математических моделях 30](#_TOC_250023)
      1. [Линеаризация произведения переменных 30](#_TOC_250022)
      2. [Линеаризация заменой переменных 32](#_TOC_250021)
      3. [Линеаризация кусочно-линейной функции 36](#_TOC_250020)
   7. [Симметрия в математических моделях 37](#_TOC_250019)
3. [Примеры математических моделей целочисленного линейного программирования 41](#_TOC_250018)
   1. [Задача о потоке минимальной стоимости 41](#_TOC_250017)
   2. [Задача коммивояжера 43](#_TOC_250016)
   3. [Задача о покрытии 46](#_TOC_250015)
   4. [Задача о двухстадийном гильотинном раскрое 48](#_TOC_250014)
   5. [Задача о разрезе балок 51](#_TOC_250013)
   6. [Задача о башнях 53](#_TOC_250012)
4. [Анализ качества моделей целочисленного линейного програм- мирования 56](#_TOC_250011)
   1. [Классификация моделей 56](#_TOC_250010)
   2. [Разрыв целочисленности 58](#_TOC_250009)
   3. [Число ограничений и переменных в модели 65](#_TOC_250008)
   4. [Многогранники. Правильные неравенства 68](#_TOC_250007)
   5. Целочисленные решения задачи линейного программирования . . 74
   6. [Уточнение значения границ переменных 77](#_TOC_250006)
   7. [Удаление избыточных ограничений 79](#_TOC_250005)
5. [Упражнения 81](#_TOC_250004)
   1. [Теоретические задания 81](#_TOC_250003)
   2. [Практические задания 92](#_TOC_250002)
6. [Решение оптимизационных задач в GAMS 123](#_TOC_250001)
7. [Список литературы 129](#_TOC_250000)

3

# Введение

Ежедневно люди принимают тысячи решений, отвечая на различные вопро- сы, начиная от простых «Где пообедать?» до сложных «В каких городах разме- стить филиалы компании?» Часто процесс принятия решения можно описать аналитически. Предположим, Вы хотите оценить время, затрачиваемое на до- рогу от работы до дома, для этого нужно узнать расстояние и разделить его на среднюю скорость передвижения. Получаем *количественную модель*:

*S*

*t* = *, v*

где *S* — расстояние, *v* — средняя скорость, *t* — время в пути. Эта модель по- лезна, но, как и любая модель, обладает существенным недостатком: упрощает и идеализирует реальность. Например, в модели не учитывается, что, возвра- щаясь с работы домой, Вы делаете остановки, чтобы зайти в магазины. Это можно учесть, добавив слагаемое:

*S*

*t* = + *nR, v*

где *n* — планируемое число остановок, *R* — среднее время на остановку. В дан- ной модели имеются постоянные величины: расстояние между домом и офисом; и переменные величины, значениями которых можно управлять: скорость пе- редвижения, число остановок и время посещения магазина. С помощью управ- ляемых переменных можно сократить затрачиваемое время на дорогу. Моде- ли, имеющие переменные величины, при изменении значений которых можно моделировать и получать разные решения, называют *моделями принятия ре- шений*, а сами переменные называют *переменными принятия решений*. Цель таких моделей не просто вычислить значение какой-то переменной, а найти ми- нимум или максимум какой-либо функции от переменных принятия решений, например, минимизировать потраченное время, максимизировать прибыль и т. д.

Рассмотрим следующий пример. Фирма производит шесть типов стульев: *капитан*, *помощник*, *маркиза*, *испанский*, *венский* и *офисный*. Для производ- ства стульев необходимы универсальные детали: длинные и короткие болты, тяжелые и легкие сиденья, длинные и короткие ножки, перекладины, гайки, роллеры, каркас и крепления. В табл. 1 приводятся данные о стоимости сту- льев, потребности в деталях для каждого типа стульев и их наличие на складе. Благодаря универсальности деталей производитель может быстро реагировать на изменения в спросе. Поступил заказ изготовить 40 стульев каждого типа из имеющихся на складе деталей.

4

*Таблица 1*

Исходные данные задачи

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тип стула *Офис. Пом. Кап. Марк. Исп. Вен.* | | | | | | | | | | |
| Цена | 36 |  | 40 |  | 45 |  | 38 | | 35 | 25 |
| Требуемое количество деталей | | | | | | | | Всего на складе | | |
| *длин. болты* | 8 | 0 | 12 | 0 |  | 8 | 4 | 1280 | | |
| *корот. болты* | 4 | 12 | 0 | 12 |  | 4 | 8 | 1900 | | |
| *тяж. сиденья* | 4 | 4 | 4 | 4 |  | 4 | 4 | 1090 | | |
| *лег. сиденья* | 1 | 0 | 0 | 0 |  | 1 | 1 | 190 | | |
| *длин. ножки* | 0 | 1 | 1 | 1 |  | 0 | 0 | 170 | | |
| *корот. ножки* | 6 | 0 | 4 | 0 |  | 5 | 0 | 1000 | | |
| *перекладины* | 0 | 4 | 0 | 5 |  | 0 | 6 | 1000 | | |
| *гайки* | 1 | 0 | 0 | 0 |  | 0 | 0 | 110 | | |
| *роллеры* | 0 | 1 | 0 | 0 |  | 0 | 0 | 72 | | |
| *каркас* | 0 | 0 | 1 | 1 |  | 0 | 0 | 93 | | |
| *крепления* | 0 | 0 | 0 | 0 |  | 1 | 1 | 85 | | |

Суммарный доход от продажи стульев определяется выражением:

6

∑

*p* = 40*ci,*

*i*=1

где *i* — тип стула, *ci* — стоимость одного стула *i*-го типа. Результаты расче- тов для склада приводятся в табл. 2. Согласно этому плану доход составляет 8760 y.e., причем длинные болты израсходованы полностью. Можно ли уве- личить доход фирмы, если количество стульев в заказе будет другим? Чтобы ответить на этот вопрос, проанализируем решение. Заметим, что для стула

«капитан» длинных болтов требуется больше, чем для других стульев.

*Таблица 2*

Первый план производства

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип стула *Офис. Пом.* | | *Кап.* | *Марк.* | *Исп.* | *Вен.* |
| Количество 40 40 | | 40 | 40 | 40 | 40 |
| Суммарный доход 8760 | | | | | |
| Всего использовано | Остаток | | | | |
| *длин. болты* 1280 | 0 | | | | |
| *корот. болты* 1600 | 300 | | | | |
| *тяж. сиденья* 960 | 130 | | | | |
| *лег. сиденья* 120 | 70 | | | | |
| *длин. ножки* 120 | 50 | | | | |
| *корот. ножки* 600 | 400 | | | | |
| *перекладины* 600 | 400 | | | | |
| *гайки* 40 | 70 | | | | |
| *роллеры* 40 | 32 | | | | |
| *каркас* 80 | 13 | | | | |
| *крепления* 80 | 5 | | | | |

5

Сократим производство стульев «капитан» на 2 единицы, а на вырученные 24 длинных болта произведем 3 «офисных» стула. При этом фирма потеряет от непроизводства «капитана» 90 y.e., но заработает 108 y.e., если продаст допол- нительно три «офисных» стула. Таким образом, суммарный доход увеличится на 18 = (43 36 + 40 40 + 38 45 + 40 38 + 40 35 + 40 25 8760) y.e.

· · · · · · −

Можно ли еще увеличить доход? Проведем те же рассуждения. Для произ- водства «испанского» стула необходимо 8 длинных болтов, а для «венского» в 2 раза меньше. Значит, если вместо одного «испанского» стула будем произво- дить два «венских» стула, то доход составит не 35 y.e., а 50 y.e. Следовательно, если фирма будет производить вместо 40 «испанских» стульев 80 «венских» стульев, то ее суммарный доход увеличится на 600 = (40 36 + 40 40 + 40

· · ·

45 + 40 38 + 0 35 + 120 25 8760) y.e. Результаты расчетов по этому плану приводятся в табл. 3. Однако этот план не может быть реализован из-за недо- статка деталей на складе, о чем свидетельствуют отрицательные значения в столбце *остаток*.

· · · −

*Таблица 3*

Второй план производства

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип стула *Офис. Пом.* | | *Кап.* | *Марк.* | *Исп.* | *Вен.* |
| Количество 40 40 | | 40 | 40 | 0 | 120 |
| Суммарный доход 9360 | | | | | |
| Всего использовано | Остаток | | | | |
| *длин. болты* 1280 | 0 | | | | |
| *корот. болты* 2320 | -180 | | | | |
| *тяж. сиденья* 1200 | -30 | | | | |
| *лег. сиденья* 220 | 30 | | | | |
| *длин. ножки* 80 | 50 | | | | |
| *корот. ножки* 600 | 600 | | | | |
| *перекладины* 1080 | -80 | | | | |
| *гайки* 100 | 70 | | | | |
| *роллеры* 40 | 32 | | | | |
| *каркас* 40 | 13 | | | | |
| *крепления* 120 | -35 | | | | |

В этой задаче мы использовали количественную модель, отражающую взаимо- связь между количеством производимых стульев, количеством деталей и сум- марным доходом. Однако если фирме нужно ответить на вопрос о том, сколько произвести стульев, чтобы получить максимальный доход, то количественная модель в этом не поможет, и нужно сформулировать модель принятия реше- ний. Далее будут описаны способы построения различных моделей принятия решений, разобрано, насколько хорошей является сформулированная матема- тическая модель и как ее улучшить.

Модели принятия решений, в которых ищется максимум (минимум) некото- рой функции от переменных принятия решений при некоторых ограничениях

6

называют также *оптимизационными моделями*. Под *ограничениями* понима- ют условия, которые препятствуют получению максимального (минимального) значения целевой функции, т. е. не дают, например, сократить затраты до 0 или достигнуть бесконечно большого дохода. В задаче про стулья в качестве огра- ничений были условия на наличие деталей на складе. В других задачах огра- ничения могут возникать в результате ограниченного бюджета, ограниченно- го времени выполнения проекта, ограниченных производственных мощностей, вместимости складов, а также из-за различных требований к производственно- му процессу, допустимым объемам затрат и др.

Оптимизационные модели активно стали использоваться в годы Второй ми- ровой войны при решении военных задач. Со временем сложились научные дисциплины *исследование операций* (от англ. operations research) и *теория при- нятия решений* (от англ. management science), которые занимаются оптимиза- ционными задачами, возникающими в экономике, производстве, политике и других областях человеческой деятельности. Под исследованием операций по- нимают применение математических и количественных методов для обоснова- ния решений во всех областях целенаправленной человеческой деятельности [2]. Под теорией принятия решений понимают особый вид человеческой дея- тельности, направленный на выбор наилучшего варианта действий [11]. Четкое разделение между этими дисциплинами трудно провести, но можно сказать, что дисциплина «Теория принятия решений» оказывается шире, чем «Иссле- дование операций», и использует ее как инструмент, когда реальную задачу удается формализовать в виде математической модели и найти наилучший ва- риант решения путем математических расчетов. Однако такая формализация часто оказывается труднореализуемой из-за нескольких критериев оптимиза- ции, слабой структурированности задачи, наличия многих лиц, принимающих решения или других причин. В таких ситуациях методы исследования опе- раций носят вспомогательный характер и выступают как инструмент теории принятия решений.

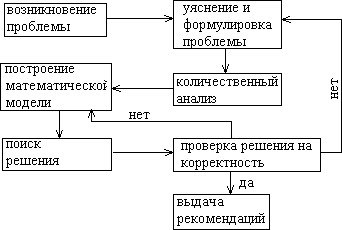
Для большинства человеческих решений нельзя точно рассчитать и оце- нить их последствия. Принимая решение при выборе того или иного вариан- та, человек интуитивно может учесть гораздо больше нюансов, чем машина. Представьте ситуацию, когда семья из нескольких человек разных поколений собирается приобрести дом. Они рассматривают несколько предложений. Каж- дый член семьи оценивает варианты домов исходя из собственных критериев. Старшие члены семьи считают, что главное — это развитая медицинская ин- фраструктура. Среднее поколение считает, что важнее всего близкое располо- жение офисов, в которые им приходится добираться каждое утро. Для млад- шего поколения (с точки зрения родителей) важнее всего наличие хорошей школы. Существуют еще и другие критерии у каждого из членов семьи: удоб- ная парковка около дома, экологическая обстановка в районе, наличие магази- нов и т.д. В подобных случаях использовать только математический аппарат для принятия решения почти невозможно, поскольку необходимо использовать

7

уникальные умения человека соизмерять противоречивые оценки, отбрасывать второстепенное, оценивать обстановку и учитывать перспективу принятого ре- шения.

Наиболее удобным и распространенным математическим инструментом при моделировании и решении оптимизационных задач является *линейное програм- мирование* — специальный класс оптимизационных задач, в котором все отно- шения между переменными выражаются линейными функциями, а перемен- ные принимают действительные значения. Преимущество этого класса в том, что разработаны универсальные алгоритмы для решения таких задач боль- шой размерности. Впервые задача линейного программирования в России была сформулирована в 1939 г. Л. В. Канторовичем, который применил математи- ческую модель этой задачи в экономике и разработал метод решения. В 1975 г. Л. В. Канторович получил Нобелевскую премию за достижения в этой области [3]. В 1947 г. американский ученый Д. Данциг разработал алгоритм решения этой задачи. С этого момента линейное программирование стало важным ин- струментом в исследовании операций.

Часто из условия задачи следует, что значения некоторых переменных при- нятия решений принадлежат множеству целых чисел. Например, если пере- менная отвечает за размер детали, то она может принимать значения толь- ко из заданного множества возможных размеров деталей. Такого рода задачи принадлежат к классу задач *целочисленного линейного программирования* или *смешанного целочисленного линейного программирования*. Они отличаются от задач линейного программирования высокой сложностью и особой практиче- ской значимостью. Задачам из этих классов в данном пособии уделяется особое внимание.



*Рис. 1.* Общая схема исследования задачи

Прежде чем переходить к построению математической модели, необходи-

8

мо пройти несколько этапов. На рис. 1 приводится общая схема исследования проблемы. Она решается в интересах так называемого лица, принимающего решения (ЛПР). ЛПР формулирует проблему, участвует в построении моде- ли, анализирует полученное решение, а затем принимает или отвергает его. Построение математической модели и решение задачи лучше доверить специ- алистам по исследованию операций.

Исследование задачи начинается с появления некоторой проблемы. Боль- шинство оптимизационных задач возникает из практических приложений.

В этом случае рекомендуется изучить процесс, в результате которого возникла задача. Может оказаться, что задача не настолько сложна, чтобы для нее ис- пользовать математический аппарат. Возможно, чтобы решить поставленную задачу, необходимо найти решение какой-то другой подзадачи, в результате которой возникла эта проблема.

Далее необходимо сформулировать задачу: определить ее цель, сформули- ровать условия, которые влияют на достижение цели. Выяснить, какие упроще- ния по сравнению с реальностью могут быть допущены в модели. Реальность сложная и многогранная, поэтому учесть все нюансы сложной прикладной за- дачи в модели не удастся. В связи с этим на этапе уяснения и формулировки задачи необходимо провести отбор условий, которые сохранят адекватность модели и не перегрузят ее несущественными деталями.

Следующий этап состоит в том, чтобы собрать все необходимые числен- ные данные. Понять, какие параметры измеряют и влияют на цель задачи, разбить их на управляемые параметры (переменные) и неуправляемые пара- метры (константы), ввести переменные, задать целевую функцию. Возможно, с первого раза не удастся наилучшим образом задать переменные. Кроме того, необходимо решить вопрос о *размерности модели*. Она определяется количе- ством переменных и ограничений. Если число управляемых параметров увели- чить, то с помощью модели можно более точно отразить реальное событие, но будет трудно выявить основные свойства модели. В такой ситуации задача ста- новится необозримой и может не иметь решения. Поэтому число переменных стараются уменьшить, оставляя главные. С другой стороны, уменьшая число переменных, можно опустить существенные переменные, и модель становится неадекватной. Большое количество ограничений тоже не всегда является недо- статком. В следующих разделах будут рассмотрены разные модели одной и той же задачи, и читатель сможет сравнить качество каждой из этих моделей.

Выполнение только первых трех этапов из общей схемы может оказаться

полезным по нескольким причинам. Во-первых, при исследовании проблемы на этапе количественного анализа нужно собрать, проверить и структуриро- вать исходные данные задачи. Например, подсчитать имеющееся количество деталей на складе, установить, какие составные части используются в произ- водстве, а какие, возможно, уже устарели, и их можно исключить из рассмот- рения. Во-вторых, определить, какие параметры и каким образом влияют на достижение цели. Выписать ограничения и тем самым наглядно показать, что

9

мешает, например, получить максимальную прибыль.

Следующий важный этап состоит в построении математической модели. Как архитектор использует бумагу для построения макета здания, так и ма- тематический язык используется для записи моделей. Может оказаться, что символическая запись слишком сложна в той постановке, которая получилась на предыдущем шаге. Сложность может заключаться в том, что критерии оп- тимизации или ограничения выражаются нелинейными функциями или число ограничений и переменных слишком велико. В результате время работы алго- ритма с такой моделью окажется неприемлемо большим. В этом случае необ- ходимо вернуться на шаг уяснения и формулировки задачи, переопределить переменные и ограничения. Построение модели — это своего рода искусство. Не существует единственного верного способа, как это делать. Для одной и той же задачи можно построить несколько эквивалентных моделей с разным чис- лом переменных и ограничений, так же как для теорем может быть предложено несколько различных доказательств.

Полезность модели зависит еще и от того, кто с ней работает. ЛПР долж- но обладать долей интуиции и понимать степень реалистичности получаемого решения. Не всегда цель моделирования состоит исключительно в том, чтобы найти *оптимальное решение задачи*, т. е. наилучшее решение для конкретной модели, но также может быть интересным исследование различных альтерна- тивных решений, с помощью которых можно прогнозировать сложные ситуа- ции. Не всегда оптимальное решение является подходящим для практического внедрения. Может оказаться, что поиск оптимума требует больших вычисли- тельных ресурсов, в то время как приближенное решение не сильно отличает- ся от оптимального или оно устраивает того, кто принимает решение. Кроме того, при моделировании часто возникают труднореализуемые аспекты. Име- ется много критериев оптимизации, и понятие оптимального решения является условным. Многовариантные расчеты по сценарию «А что, если...?» позволяют проверить чувствительность решений к изменениям исходных данных, исследо- вать различные предположения и оценить последствия принимаемых решений.

В большинстве задач выбора имеется много критериев оценки вариантов

решения. Критерии могут быть зависимыми или независимыми. Зависимы- ми называют те критерии, при которых оценка альтернативы по одному из них определяет оценку по другому критерию. Количество критериев влияет на сложность задачи принятия решений. При небольшом числе критериев (два- три) задача сравнения по ним может быть выполнена непосредственно. При большом числе критериев задача становится малообозримой, и тогда необхо- димо использовать специальные методы решения многокритериальных задач оптимизации [7, 27].

В пособии будут рассмотрены задачи, в которых поиск решения проводится по одному критерию оптимизации. Материал, содержащийся в пособии, явля- ется частью лекционных курсов и семинарских занятий по дисциплине «Тео- рия принятия решений» [1], читаемой студентам 3 и 5-го курсов на факультете

10

информационных технологий Новосибирского государственного университета. Программа этой дисциплины составлена в соответствии с требованиями ФГОС ВПО к структуре и результатам освоения основных образовательных программ магистратуры по циклу «Общие математические и естественно-научные дис- циплины» по направлению подготовки «Информатика и вычислительная тех- ника», а также задачами, стоящими перед Новосибирским государственным университетом по реализации Программы развития НГУ.

11

# Моделирование с помощью булевых переменных

В общем виде оптимизационная модель состоит из целевой функции и огра- ничений. Целевая функция устанавливает зависимость критерия оптимизации от управляемых и неуправляемых параметров *x* и *k* соответственно в виде функции *f* (*x, k*). Функция может быть задана аналитически или с помощью алгоритма. Для *f* (*x, k*) указывается, по какому критерию будет проводиться оптимизация, минимизация или максимизация:

min *f* (*x, k*) или max *f* (*x, k*)*.* (2.1)

Все ограничения модели записываются с помощью некоторых функций, также зависящих от управляемых и неуправляемых параметров *x* и *k* в виде нера- венств:

*gi*(*x, k*) ≤ 0*, i* = 1*, . . . , n.* (2.2)

Как правило, переменные в задачах принимают значения из заданного стан- дартного множества *X*. Например, из множества положительных действитель- ных чисел. Поэтому в модели возникают требования на значения управляемых параметров:

*x* ∈ *X.* (2.3)

Итак, получаем общий вид математической модели (2.1)—(2.3).

Если все взаимосвязи между управляемыми и неуправляемыми параметра- ми выражаются с помощью линейных функций, то специалист имеет дело с линейными моделями. Их основное преимущество — это возможность приме- нения линейного программирования. Современные программные средства для решения задач комбинаторной оптимизации так или иначе используют метод ветвей и границ, методы отсечений и др. Очень часто эти методы проще ре- ализуются для моделей с линейными зависимостями между параметрами и переменными.

Переменная называется *булевой* в честь английского математика Дж. Бу- ля, если она принимает одно из двух значений 0 или 1. Иногда некоторые условия легко и естественным образом могут выражаться с помощью нелиней- ных функций. В этом случае для линеаризации приходится вводить дополни- тельные переменные, значения которых полностью определяются исходными переменными принятия решений; выписывать с помощью линейных функций ограничения, связывающие дополнительные переменные с исходными. Спосо- бы линеаризации будут рассмотрены далее в разд. 2.6.

В разд. 2.2 будут представлены правила построения линейных неравенств и равенств с булевыми переменными для выражения логических взаимосвя- зей между событиями. Слово «правило» не должно вызывать противоречия со словом «искусство» применительно к моделированию, поскольку некоторые общие закономерности при моделировании все же существуют.

Прежде чем переходить к моделированию с помощью булевых переменных, заметим, что любую переменную, которая принимает целое неотрицательное

12

значение, ограниченное сверху, можно представить как сумму булевых пере- менных с соответствующими коэффициентами. Например, пусть переменная *x* принимает целые значения из отрезка [0*, U* ]. Тогда *x* можно представить, ис- пользуя двоичное разложение следующим образом:

⌊*log*2 *U* ⌋

∑

*x* = 2*jxj,*

*j*=0

где *xj* 0*,* 1 для всех *j* = 0*, . . . , log*2*U* .

∈ { } ⌊ ⌋

Кроме построения модели, может возникнуть обратная задача, в которой

по математической модели необходимо понять содержательный смысл ограни- чений и взаимосвязей между переменными. Например, пусть имеются булевы переменные *x*1, *x*2, *x*3, непрерывная переменная *y*, принимающая значения из отрезка [0*,* 1], и неравенство

*x*1 − *x*2 + *x*3 + *y* ≤ 2*.*

Возникает вопрос, какие взаимосвязи между переменными реализованы в этом неравенстве? После недолгих размышлений нетрудно заметить, что это нера- венство моделирует следующую зависимость: если *x*1 = 1, *x*2 = 0, *x*3 = 1, то *y* 0, учитывая, что *y* [0*,* 1], следует, что *y* = 0. Моделирует ли это неравен- ство еще какие-либо зависимости между этими переменными? Ответ «нет», а почему это так, будет объясняться ниже.

≤ ∈

## Примеры математических моделей

Прежде чем переходить к специальным приемам и разным способам моде- лирования, рассмотрим несколько простых с точки зрения построения матема- тических моделей примеров хорошо известных классических задач.

*Задача о 0–1 рюкзаке*

Имеется *n* предметов и рюкзак грузоподъемностью *V* . Известна полезность *cj* и вес *vj* каждого предмета, *j* = 1*, . . . , n*. Требуется определить, какие пред- меты положить в рюкзак, чтобы суммарная полезность выбранных предметов была наибольшей, а общий вес не превышал грузоподъемность рюкзака [29].

Для моделирования событий «предмет положили в рюкзак» или «не поло- жили» введем *n* булевых переменных:

{*x* =*j*

1*,* если предмет *j* положили в рюкзак,

0*,* если предмет *j* не положили в рюкзак.

Целевая функция задачи — максимальная суммарная полезность выбранных предметов:

∑

*n*

max *cjxj.* (2.4)

*j*=1

13

Тому, чтобы положить все предметы в рюкзак и таким образом достигнуть максимальной полезности, препятствует ограничение на общий вес выбранных предметов. Он не должен превышать грузоподъемность рюкзака:

*n*

∑

*vjxj* ≤ *V.* (2.5)

*j*=1

Ограничения на значения переменных:

*xj* ∈ {0*,* 1}*, j* = 1*, . . . , n.* (2.6)

Итак, получили математическую модель (2.4)–(2.6) для задачи о 0–1 рюкзаке с *n* булевыми переменными и одним ограничением.

*Задача о назначениях*

Имеется *n* рабочих и *m* работ, *n m*. Известны (*cij*) затраты при выпол- нении работы *j* рабочим *i*. Требуется составить план назначений рабочих на работы так, чтобы все работы были выполнены с минимальными суммарными затратами.

≥

Введем *nm* булевых переменных:

{*x* =*ij*

1*,* если рабочий *i* выполняет работу *j,*

0− в противном случае.

Целевая функция — минимальные суммарные затраты:

*n m*

∑ ∑

min *cijxij.* (2.7)

*i*=1 *j*=1

При условиях, что каждая работа должна быть выполнена

*n*

∑

*xij* = 1*, j* = 1*, . . . , m,* (2.8)

*i*=1

каждый рабочий может выполнять не более одной работы

*m*

∑

*xij* ≤ 1*, i* = 1*, . . . , n* (2.9)

*j*=1

и при ограничениях на значения переменных

*xij* ∈ {0*,* 1}*, i* = 1*, . . . , n, j* = 1*, . . . , m.* (2.10)

В литературе можно встретить разные интерпретации этой задачи. Она по- линомиально разрешима, в [1, 13] для случая с *n* = *m* описывается алгоритм, трудоемкость которого составляет *O*(*n*3).

14

*Задача о размещении студентов в общежитии*

Рассмотрим задачу, близкую к задаче о назначениях. Имеется 2*n* студентов, которых нужно расселить по *n* двухместным комнатам, т. е. каждому студенту назначить ровно одного соседа. Величина (*cij*) задает расстояние между горо- дами, в которых жили студенты *i* и *j* до поступления в университет. Требуется разместить студентов по комнатам таким образом, чтобы перезнакомить как можно больше студентов из наиболее удаленных друг от друга городов.

Введем булевы переменные:

{*x* =*ij*

1*,* если студенты *i* и *j* живут в одной комнате,

0 − в противном случае.

Целевая функция — максимальное расстояние между познакомившимися сту- дентами

2*n*−1

∑

max

2*n*

*cijxij,* (2.11)

∑

*i*=1 *j*=*i*+1

при ограничениях, что у каждого студента ровно один сосед

∑ *xki* + ∑ *xij* = 1*, i* = 1*, . . . ,* 2*n,* (2.12)

*k<i*

*j>i*

*k<i*

*j>i*

и ограничениях на значения переменных

*xij* ∈ {0*,* 1}*, i* = 1*, . . . ,* 2*n, j* = 1*, . . . ,* 2*n.* (2.13)

В литературе задача (2.11)–(2.13) известна как *задача о совершенном па- росочетании* [8, 13]. *Паросочетанием* в графе *tt* с множеством вершин *V* и множеством ребер *E* называется подмножество ребер *M E*, такое, что для всех вершин *v* из *V* в *M* содержится не более одного ребра, инцидентного *v*. Вершина *v* из *V* называется *связанной паросочетанием*, если в *M* есть ребро, инцидентное *v*. *Совершенное паросочетание* — это паросочетание, в котором каждая вершина является связанной.

⊆

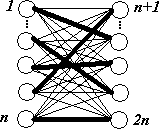
В задаче (2.11)–(2.13) требовалось найти совершенное паросочетание мак- симального веса на двудольном графе (рис. 2).

Если в (2.12) знак равенства заменить на неравенство меньше либо рав- но, то получим задачу о поиске паросочетания максимального веса. Задача о назначениях с *n > m*, рассмотренная выше, эквивалентна задаче о поиске па- росочетания минимального веса на двудольном графе (рис. 3).

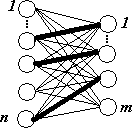
## Правила моделирования логических импликаций

Рассмотрим некоторые правила, следуя которым можно моделировать логи- ческие импликации с помощью линейных ограничений и булевых переменных [26].

15



*Рис. 2.* Cовершенное паросочетание для задачи о размещении студентов



*Рис. 3.* Паросочетание минимального веса для задачи о назначениях

Напомним, что под *импликацией* понимается логическая связка некоторого условия и следствия из него и выражается союзами «если..., то...».

***Первое правило.*** Пусть *I* — конечное множество индексов, *xi*— булева переменная, *i I* и *y* — непрерывная переменная, такая, что 0 *y* 1. Тогда импликация

∈ ≤ ≤

если *xi* = 0 для всех *i* ∈ *I,* то *y* = 0

моделируется неравенством

∑

*y* ≤ *xi.* (2.14)

*i*∈*I*

Нетрудно проверить, что неравенство (2.14) действительно реализует нужную логическую связку. Если *xi* = 0 для всех *i I*, то *i I xi* = 0 и неравен- ство (2.14) превращается в неравенство *y* 0. Поскольку *y* — неотрицательная величина, то *y* = 0.

∑ ∈

≤

∈

Более того, неравенство (2.14) не порождает никаких лишних ограничений, т. е. если *xi* = 1 для некоторого *i*, то 1 ≤ *i*∈*I xi*, и так как *y* ≤ 1, то неравенство

∑

16

(2.14) всегда выполнено.

*Следствие 1.1.* Пусть значение переменной *y* ограниченно сверху величиной

*c*, 0 ≤ *y* ≤ *c*. Тогда импликация

если *xi* = 0 для всех *i* ∈ *I,* то *y* = 0

моделируется неравенством

∑

*y* ≤ *c xi.* (2.15)

*i*∈*I*

*Пример. Задача размещения производства* [9]

Задано конечное множество возможных мест производства *I* некоторой од- нородной продукции и конечное множество клиентов *J* . Известны затраты *ci* на организацию производства в пункте *i*. Продукция доставляется клиентам, сто- имость доставки клиенту *j* из пункта *i* равна *dij*. Каждый клиент может обслу- живаться только из одного пункта. Необходимо определить, в каких пунктах следует разместить производство, чтобы обслужить всех клиентов с наимень- шими суммарными затратами.

Введем булевы переменные:

{*x* =*i*

1*,* если в пункте *i* размещается производство,

0 − в противном случае

и



*yij* =

1*,* если клиент *j* обслуживается из пункта производства *i,*

 0 − в противном случае.

Первая группа ограничений должна гарантировать, что каждый клиент будет

обслуживаться ровно одним открытым предприятием:

∑

*yij* = 1*,* для каждого *j* ∈ *J.* (2.16)

*i*∈*I*

Вторая группа ограничений гарантирует, что если в пункте *i* производство не размещено, то клиент *j* из него не обслуживается. В виде импликации это условие записывается так:

если *xi* = 0*,* то *yij* = 0*,* для каждого *i* ∈ *I, j* ∈ *J .*

Следуя первому правилу, получаем:

*yij* ≤ *xi,* для каждого *i* ∈ *I, j* ∈ *J.* (2.17)

Выпишем целевую функцию. Для этого вычислим суммарные затраты, кото- рые нужно минимизировать. Они складываются из затрат на организацию про- изводства и суммарных затрат на обслуживание клиентов из открытых пунк- тов производства:

min ∑ *cixi* + ∑ ∑ *dijyij.* (2.18)

*i*∈*I*

*i*∈*I j*∈*J*

17

Получили математическую модель (2.16)–(2.18) с булевыми переменными *xi* и

*yij*.

На практике предприятия не могут производить, а клиенты не могут по- треблять бесконечное количество продукции. Поэтому более реалистичная си- туация, когда производственные мощности предприятий и спрос у клиентов ограничены.

*Задача размещения с ограничениями на мощности производства* [9]

Пусть производственная мощность предприятия *i* составляет *ui* единиц, а величина *bj* задает спрос клиента *j*. Величины *dij* задают удельные затраты на доставку продукции от клиента *j* до пункта *i*. Изменим смысл переменных *yij*. Теперь они будут обозначать количество продукции, поставляемое клиенту *j* из пункта *i*. Тогда задача размещения с ограничениями на мощности производства может быть записана в следующем виде:

целевая функция

min ∑ *cixi* + ∑ ∑ *dijyij,* (2.19)

*i*∈*I*

*i*∈*I j*∈*J*

ограничения, гарантирующие удовлетворение спроса для каждого клиента,

∑

*yij* = *bj,* для каждого *j* ∈ *J.* (2.20)

*i*∈*I*

В модели не хватает ограничений, которые гарантировали бы, что если в пункте *i* открыто предприятие, то общее количество продукции, отправленное из него всем клиентам, не может быть больше, чем величина *ui*, т. e.

если *xi* = 0*,* то *yij* = 0*,* для каждого *j* ∈ *J*

∑≤ ≤∈

и 0 *j J yij ui*. Модифицируя первое правило, эти условия моделируются неравенствами:

∑

*yij* ≤ *uixi,* для каждого *i* ∈ *I.* (2.21)

*j*∈*J*

Добавив ограничения на значения переменных:

*xi* ∈ {0*,* 1}*, i* ∈ *I* (2.22)

*yij* ≥ 0*, i* ∈ *I, j* ∈ *J,* (2.23)

получаем математическую модель (2.19)–(2.23) смешанного целочисленного ли- нейного программирования.

*Пример. Задача о покрытии*

Задано конечное множество клиентов *J* и конечное множество пунктов раз- мещения магазинов *I*. Известны кратчайшие расстояния *dij* между элементами *i*, *j* из множеств *I* и *J* соответственно. Величина *sj* задает максимальное рас- стояние, которое клиент *j* согласен преодолеть до магазина. Учитывая это, для

18

каждого клиента *j* можно определить множество магазинов *Nj*, которые он мог бы посещать: *Nj* := *i I dij sj* .

{ ∈ | ≤ }

Требуется определить минимальное количество магазинов, которое нужно открыть, чтобы обслужить всех клиентов. Другими словами, требуется вы- брать минимальное по числу элементов подмножество множества *I*, покрыва- ющее все множество *J* .

Введем булевы переменные:

{*x* =*i*

1*,* если в *i*-м пункте магазин открыт,

0 − в противном случае.

Тогда целевая функция — минимизировать общее число открываемых пред- приятий:

∑

min *xi.* (2.24)

*i*∈*I*

Ограничения, гарантирующие, что каждый клиент будет посещать подходящий магазин:

∑

*xi* ≥ 1*, j* ∈ *J.* (2.25)

*i*∈*Nj*

Ограничения на значения переменных:

*xi* ∈ {0*,* 1}*, i* ∈ *I.* (2.26)

Получаем математическую модель (2.24)–(2.26) для одной из задач о покрытии. Существуют и другие постановки задач о покрытии, которые образуют целый класс этих задач [21, 30].

В некоторых задачах из этого класса покрытие каждого объекта связано с определенными затратами, а общий бюджет ограничен, поэтому покрыть мно- жество *J* полностью невозможно. В этом случае решают задачу о поиске мак- симального количества покрытых объектов.

Пусть *wi* — затраты, связанные с открытием магазина в пункте *i*, *B* — об- щий бюджет на открытие магазинов. Требуется обслужить как можно больше клиентов при ограниченном бюджете на открытие магазинов.

Введем новые булевы переменные:

*yj* = {

1*,* если клиент *j* обслуживается,

0 − в противном случае.

Целевая функция — максимальное число обслуживаемых клиентов:

∑

max *yj.* (2.27)

*j*∈*J*

Переменные *yj* и *xi* логически связаны между собой следующим образом:

*yj* = 1*,* тогда и только тогда, когда *xi* = 1 для некоторого *i* ∈ *Nj.*

19

Это условие содержит две импликации. Смоделируем отдельно каждую из них. Первая импликация:

если *xi* = 1 для некоторого *i* ∈ *Nj,* то *yj* = 1

или

если *xi* = 0 для всех *i* ∈ *Nj,* то *yj* = 0*.*

Следуя первому правилу, получаем:

∑

*yj* ≤ *xi j* ∈ *J.* (2.28)

*i*∈*Nj*

Вторая импликация:

если *yj* = 1*,* то *xi* = 1 для некоторого *i* ∈ *Nj*

или

если *yj* = 0*,* то *xi* = 0 для всех *i* ∈ *Nj*

Применяя первое правило, получаем формулировку этой импликации в виде следующего неравенства:

*xi* ≤ *yj,* для всех *i* ∈ *Nj, j* ∈ *J.* (2.29) Суммарные затраты на открытие магазинов ограничены:

∑

*wixi* ≤ *B.* (2.30)

*i*∈*I*

Ограничения на значения переменных:

*xi, yj* ∈ {0*,* 1}*, i* ∈ *I, j* ∈ *J.* (2.31)

***Второе правило.*** Пусть *xi* — булева переменная, *i I*, где *I* — конечное множество индексов; *I*0 и *I*1 — непересекающиеся подмножества множества *I* и *y* — целочисленная, или непрерывная, переменная, удовлетворяющая неравен- ству 0 ≤ *y* ≤ 1. Тогда логическая импликация

∈

если *xi* = 0 для всех *i* ∈ *I*0 и *xi* = 1 для всех *i* ∈ *I*1*,* то *y* = 0

моделируется неравенством

*y* ≤ ∑ *xi* + ∑(1 − *xi*)*.* (2.32)

*i*∈*I*0 *i*∈*I*1

Логическая импликация

если *xi* = 0 для всех *i* ∈ *I*0 и *xi* = 1 для всех *i* ∈ *I*1*,* то *y* = 1

20

моделируется неравенством

(1 − *y*) ≤ ∑ *xi* + ∑(1 − *xi*)*.* (2.33)

*i*∈*I*0

*i*∈*I*1

*i*∈*I*0

*i*∈*I*1

Эквивалентность импликаций и неравенств нетрудно проверить самостоя- тельно, пользуясь первым правилом.

После приведения подобных неравенства (2.32) и (2.33) можно переписать

в виде

∑ *xi* − ∑ *xi* + *y* ≤ |*I*1| (2.34)

*i*∈*I*1

и

*i*∈*I*0

∑ *xi* − ∑ *xi* − *y* ≤ |*I*1| − 1*.* (2.35)

*i*∈*I*1

*i*∈*I*0

*i*∈*I*1

*i*∈*I*0

Этот вид более компактный, но по нему сложнее восстановить логические

взаимосвязи между переменными.

Итак, *если импликация содержит xi* = 1 *для некоторого i, то в неравен- ство вместо переменной xi будет входить* (1 *xi*)*.*

−

*Пример*

Смоделируем импликацию:

если *x*1 = 1 и *x*2 = 0 и *x*3 = 1*,* то *y* = 0*.*

Следуя второму правилу, получаем:

*y* ≤ (1 − *x*1) + *x*2 + (1 − *x*3) (2.36)

или

*x*1 − *x*2 + *x*3 + *y* ≤ 2*.*

Это неравенство рассматривалось в разд. 2, и теперь должно быть понятно, какие взаимосвязи оно реализует.

*Пример. Выбор ближайшего открытого предприятия*

Как правило, в задачах размещения предприятий среди всех открытых предприятий клиенты выбирают для себя ближайшее. Иногда такие назначе- ния происходят автоматически, если целевая функция направлена на миними- зацию суммарного расстояния пройденного клиентами. Однако если целевая функция другая или в задаче присутствует ограничение, например, на бюджет проекта, то в оптимальном решении клиенты не обязательно будут назначены к ближайшим предприятиям. И тогда возникает необходимость моделировать в виде ограничения условие: *если в пункте s размещено предприятие и другие открытые предприятия дальше, чем s по отношению к клиенту i, то клиент i должен обслуживаться из предприятия s*.

21

Для моделирования введем следующие булевы переменные:

{*x* =*s*

1*,* если предприятие размещено в пункте *s,*

0 − в противном случае,

{*y* =*is*

1*,* если клиент *i* назначен к предприятию *s,*

0 − в противном случае.

Обозначим через *Cis* множество всех предприятий, которые находятся к кли- енту *i* ближе, чем предприятие *s*. Тогда

если *xs* = 1 и *xt* = 0 для всех *t* ∈ *Cis,* то *yis* = 1*.*

Применяя второе правило, получаем:

∑

1 − *yis* ≤ (1 − *xs*) + *xt*

*t*∈*Cis*

или

*yis* ≥ *xs* − *xt.*

∑

*t*∈*Cis*

## Моделирование свойств логических отношений

***Моделирование отношения транзитивности.*** Часто в задачах меж- ду объектами заданного множества *I* определено некоторое отношение *R*. На- пример, *i < j*, в данном случае под *R* понимается отношение сравнения «строго меньше». Как правило, *R* должно обладать свойством транзитивности, т. е. ес- ли *iRj* и *jRk*, то *iRk* для любых *i, j, k I.*

∈

Рассмотрим граф с множеством вершин *V* и множеством ребер *E*. Требует- ся смоделировать транзитивность отношения связности вершин. Пусть булева переменная:

{*x* =*ij*

1*,* если между вершинами *i* и *j* есть путь,

0 − в противном случае.

Тогда для любой тройки вершин (*i, j, k*) ∈ *V* × *V* × *V*

если *xij* = 1 и *xjk* = 1*,* то *xik* = 1*.*

Следуя второму правилу, получаем линейное неравенство:

(1 − *xik*) ≤ (1 − *xij*) + (1 − *xjk*)

или

*xij* + *xjk* − *xik* ≤ 1*.* (2.37)

22

***Моделирование отношения порядка.*** В задачах теории расписаний между работами, выполняемыми на одной машине, должны соблюдаться от- ношения предшествования. Очередная работа не может начаться, пока не за- кончится предыдущая работа. Отношения порядка, т. е. транзитивности и ан- тисимметричности на множестве работ, в таких задачах можно моделировать по-разному. Рассмотрим вариант со следующими переменными.

Введем булевы переменные:

{*x* =*ij*

1*,* если работа *i* выполняется до работы *j, i* ̸= *j*

0 − в противном случае.

Запишем условие антисимметричности. Для любой пары работ *i* и *j* всегда одна работа должна выполняться раньше другой, т. е. только одна из переменных *xij* или *xji* может быть равна единице:

*xij* + *xji* = 1*,*

и *xii* = 0*.* Следовательно, можно ограничиться рассмотрением только перемен- ных *xij* с *i < j* и *xii* = 0.

Для каждой тройки работ *i*, *j* и *k* должно выполняться отношение транзи- тивности:

при *i < j < k* (1 − *xik*) ≤ (1 − *xij*) + (1 − *xjk*)*,* при *i < k < j* (1 − *xij*) ≤ (1 − *xik*) + (1 − *xkj*)*,* при *k < i < j* (1 − *xkj*) ≤ (1 − *xki*) + (1 − *xij*)*,* при *k < j < i* (1 − *xki*) ≤ (1 − *xkj*) + (1 − *xji*)*,* при *j < i < k* (1 − *xjk*) ≤ (1 − *xji*) + (1 − *xik*)*,* при *j < k < i* (1 − *xji*) ≤ (1 − *xjk*) + (1 − *xki*)*.*

С учетом антисимметричности, при переходе к переменным *xij* с *i < j* из шести групп неравенств останутся только следующие неравенства:

*xij* + *xjk* − *xik* ≤ 1 для любых (*i, j, k*)*, i < j < k,* (2.38)

(1 − *xij*) − *xjk* + *xik* ≤ 1 для любых (*i, j, k*)*, i < j < k.* (2.39)

Неравенство (2.38) гарантирует транзитивность в очередности выполнения ра- бот *i, j* и *k*, если работа *i* предшествует работе *j* (*xij* = 1). Если это не так, то неравенство (2.38) верно при любых значениях *xjk* и *xik*, а с помощью неравен- ства (2.39) сохраняется транзитивность при *xij* = 0.

***Моделирование отношений эквивалентности.*** В отличие от преды- дущего примера, в некоторых задачах требуется, чтобы между объектами со- хранялось отношение эквивалентности, т. е. соблюдалась и транзитивность, и симметричность, например, в задачах кластеризации, в которых некоторое ко- нечное множество объектов *I* нужно разбить на подмножества. Пусть булевы

23

переменные *xij* показывают, принадлежат ли объекты *i* и *j* одному подмноже- ству:

1*,* если объекты *i* и *j* лежат в одном подмножестве,

{*x* =*ij*

0 − в противном случае.

В силу симметричности принадлежности двух объектов одному подмножеству число переменных можно сократить и рассматривать только *xij* с *i < j*, доба- вив условия, что *xji* = *xij*, и условие рефлексивности *xii* = 1. Кроме того, для любой тройки должно выполняться отношение транзитивности. Но, в отличие от предыдущего примера из теории расписаний, в задачах разбиения между объектами, попавшими в одно подмножество, должна соблюдаться симметрич- ность. Поэтому кроме одного неравенства типа (2.37) нужно сформулировать еще два неравенства. Итак, первое неравенство гарантирует, что если *i* и *j* в одном подмножестве, *j* и *k* в одном подмножестве, то *i* и *k* также в одном подмножестве, т. е.

*xij* + *xjk* − *xik* ≤ 1 для любых (*i, j, k*)*, i < j < k.* (2.40)

Второе неравенство гарантирует, что если *i* и *j* в одном подмножестве, *i* и *k* в одном подмножестве, то *j* и *k* также в одном подмножестве, т. е.

*xij* + *xik* − *xjk* ≤ 1 для любых (*i, j, k*)*, i < j < k.* (2.41)

Третье неравенство гарантирует, что если *i* и *k* в одном подмножестве, *j* и *k* в одном подмножестве, то *i* и *j* также в одном подмножестве, т. е.

*xik* + *xjk* − *xij* ≤ 1 для любых (*i, j, k*)*, i < j < k.* (2.42)

## Моделирование выбора минимального элемента

Часто в задачах нужно, чтобы переменная из нескольких значений при- нимала минимальное значение. Например, необходимо, чтобы переменная *y* принимала значение, равное min(*u*1*, u*2), где *u*1*, u*2 0. Значит, одновременно должны выполняться неравенства

≥

*y* ≤ *u*1 и *y* ≤ *u*2

и одно из неравенств

*y* ≥ *u*1 или *y* ≥ *u*2*.*

Сложность возникает в моделировании выполнения одного из двух последних неравенств. Введем две вспомогательные булевы переменные:

1*,* если неравенство *y* ≥ *u*1 выполняется,

{*x* =1

0 − в противном случае,

(2.43)

24

1*,* если неравенство *y* ≥ *u*2 выполняется,

{*x* =2

0 − в противном случае,

(2.44)

через *W* обозначим некоторое большое положительное число. Тогда выбор ми- нимального из двух чисел *u*1 и *u*2 эквивалентен совместности следующей си- стемы ограничений:

|  |  |
| --- | --- |
| *y* ≤ *u*1 | (2.45) |
| *y* ≤ *u*2 | (2.46) |
| *x*1 + *x*2 = 1 | (2.47) |
| *y* ≥ *u*1 − *W* (1 − *x*1) | (2.48) |
| *y* ≥ *u*2 − *W* (1 − *x*2) | (2.49) |

Ограничение (2.47) говорит о том, что ровно одна из переменных, *x*1 или *x*2, будет равна 1. Предположим, что *u*1 *< u*2. Тогда случай, когда *x*1 = 0, *x*2 = 1, не возможен потому, что ограничение (2.48) будет выполнено при любом значении *y*, но из ограничения (2.49) получается *y u*2, следовательно, система (2.45)– (2.49) несовместна. Остается вариант, когда *x*1 = 1, *x*2 = 0, в этом случае условие (2.49) будет выполнено при любом значении *y*, а согласно неравенствам (2.45) и (2.48) получаем *y* = *u*1.

≥

## Моделирование взаимоисключающих событий

Рассмотрим ситуацию, в которой возникает необходимость выполнения неко- торой подгруппы ограничений. Предположим, что допустимая область форми- руется из нескольких групп неравенств. Например, пусть допустимая область соответствует заштрихованной части, как показано на рис. 4, и образована объ- единением двух многоугольников *P*1 и *P*2.

Каждый многоугольник задается группой неравенств.

*P*1 :

*P*2 :

2*y*1 + *y*2 ≥ 4*,*

*y*1 − *y*2 ≥ −4*,*

−4*y*1 + 3*y*2 ≥ −8*,*

2*y*1 + 3*y*2 ≤ 22*,*

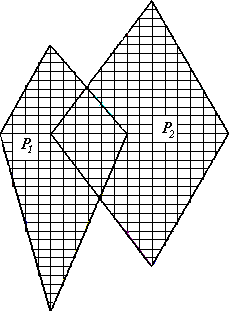
*y*1 ≥ 0*, y*2 ≥ 0*.*

3*y*1 + 4*y*2 ≥ 22*,*

−3*y*1 + 4*y*2 ≤ 10*,* 3*y*1 + 3*y*2 ≤ 39*,*

*y*1 − *y*2 ≤ 5*,*

25



*Рис. 4.* Допустимая область

*y*1 ≥ 0*, y*2 ≥ 0*.*

Решение является допустимым, если оно удовлетворяет хотя бы одной груп- пе неравенств *P*1 или *P*2. Другими словами, из двух групп ограничений нужно, чтобы выполнялась, по крайней мере, одна группа. Рассмотрим два способа моделирования этой ситуации.

*Первый способ*

Преобразуем все ограничения, которые этого требуют, кроме ограничений на значение переменных, в ограничения со знаком меньше либо равно в нера- венствах:

*P*1 :

*P*2 :

−2*y*1 − *y*2 ≤ −4*,*

−*y*1 + *y*2 ≤ 4*,* 4*y*1 − 3*y*2 ≤ 8*,*

2*y*1 + 3*y*2 ≤ 22*,*

*y*1 ≥ 0*, y*2 ≥ 0*.*

−3*y*1 − 4*y*2 ≤ −22*,*

−3*y*1 + 4*y*2 ≤ 10*,* 3*y*1 + 3*y*2 ≤ 39*,*

26

Введем булевы переменные:

*y*1 − *y*2 ≤ 5*, y*1 ≥ 0*, y*2 ≥ 0*.*

1*,* если выполнена группа неравенств для *P*1*,*

{*x* =1

0 − в противном случае,

1*,* если выполнена группа неравенств для *P*2*,*

{*x* =2

0 − в противном случае.

(2.50)

(2.51)

Введем вспомогательный вектор (*w*1*, w*2*, w*3*, w*4) с числом компонент, рав- ным максимальному числу неравенств в *P*1 и *P*2, а значения компонент — боль- шие положительные числа, такие, что при добавлении компоненты к правой части соответствующего неравенства это неравенство выполняется при любых значениях *y*1 и *y*2. Выпишем следующую систему ограничений *I*:

−2*y*1 − *y*2 ≤ −4 + *w*1(1 − *x*1)*,*

−*y*1 + *y*2 ≤ 4 + *w*2(1 − *x*1)*,* 4*y*1 − 3*y*2 ≤ 8 + *w*3(1 − *x*1)*,*

2*y*1 + 3*y*2 ≤ 22 + *w*4(1 − *x*1)*,*

−3*y*1 − 4*y*2 ≤ −22 + *w*1(1 − *x*2)*,*

−3*y*1 + 4*y*2 ≤ 10 + *w*2(1 − *x*2)*,* 3*y*1 + 3*y*2 ≤ 39 + *w*3(1 − *x*2)*, y*1 − *y*2 ≤ 5 + *w*4(1 − *x*2)*,*

*x*1 + *x*2 ≥ 1*,* (2.52)

*x*1*, x*2 ∈ {0*,* 1}*,*

*y*1 ≥ 0*, y*2 ≥ 0*.*

Условие (2.52) означает, что, по крайней мере, одна группа ограничений для *P*1 или для *P*2 выполняется. Если *x*1 = 0, то в силу (2.52) получается *x*2 = 1, и группа ограничений для *P*1 становится избыточной из-за наличия компонент вспомогательного вектора, а выполняться должна другая группа ограничений для *P*2. Аналогично если *x*2 = 0, то *x*1 = 1, и решение должно удовлетворять условиям для *P*2. Если *x*1 = 1, *x*2 = 1, то обе группы ограничений должны быть выполнены.

*Второй способ*

27

Увеличим число переменных следующим образом. Будем разделять пере- менные *y*1 и *y*2 на два типа. В системе неравенств, определяющей область *P*1, бу- дем использовать переменные *y*1, *y*1. В системе неравенств, задающих область

1 2

*P*2, будем использовать переменные *y*2, *y*2. Причем *y*1 + *y*2 = *y*1, *y*1 + *y*2 = *y*2.

1 2 1 1 2 2

Кроме того, введем булевы переменные *x*1, *x*2, смысл которых, как и в первом способе. Запишем систему ограничений *II* :

2*y*1 + *y*1 ≥ 4*x*1*,*

1

2

*y*1 − *y*1 ≥ −4*x*1*,*

1 2

1 1

−4*y*1 + 3*y*2 ≥ −8*x*1*,*

2*y*1 + 3*y*1 ≤ 22*x*1*,*

1

2

3*y*2 + 4*y*2 ≥ 22*x*2*,*

1 2

2 2

−3*y*1 + 4*y*2 ≤ 10*x*2*,*

3*y*2 + 3*y*2 ≤ 39*x*2*,*

1

2

*y*2 − *y*2 ≤ 5*x*2*,*

1

2

*y*1 + *y*2 = *y*1*,*

1 1

*y*1 + *y*2 = *y*2*,*

2 2

*x*1 + *x*2 1*,* (2.53)

≥

*y*1 ≥ 0*, y*1 ≥ 0*, y*2 ≥ 0*, y*2 ≥ 0*, y*1 ≥ 0*, y*2 ≥ 0*,*

1

2

1

2

*x*1*, x*2 ∈ {0*,* 1}*.*

Утверждается, что *P*1 *P*2 = Ø тогда и только тогда, когда система *II*

∪ ̸

разрешима. Проверим это.

Пусть *y*1, *y*2 (*P*1 *P*2), можно считать, что *y*1, *y*2 *P*1. Тогда полагаем

∈ ∪ ∈

*x*1 = 1, *x*2 = 0, *y*1 = *y*1, *y*2 = *y*1, *y*2 = *y*2 = 0 — решение системы.

1 2 1 2

Пусть, без ограничений общности, *x*1 = 1, *x*2 = 0, *y*1 = *y*1, *y*2 = *y*1, *y*2 = *y*2 =

1 2 1 2

0 — решение системы *II*. Тогда *y*1*, y*2 *P*1, следовательно, *P*1 *P*2 = Ø.

∈ ∪ ̸

К вопросу о том, какой из двух способов лучше, вернемся позже, а пока под-

черкнем, что этот пример показывает, как с помощью линейных ограничений и неравенств моделировать взаимоисключающие условия. Такие условия часто возникают в задачах составления расписаний. Например, если нужно задать порядок выполнения работ, в котором выполнение некоторой работы нельзя начать, пока не закончится другая работа.

*Задача составления расписания*

Имеется *n* работ и *m* машин. Для каждой работы задан порядок выпол- нения на машинах, т. е. работа *j* сначала выполняется на машине с номером

28

*j*(1), затем на машине с номером *j*(2) и т. д. В каждый момент времени ма- шина может выполнять не более одной работы, каждая работа выполняется не более, чем на одной машине. Работы не прерываются. Известна длитель- ность *pij* выполнения работы *j* на машине *i*. Требуется выполнить все работы и минимизировать сумму времен завершения всех работ.

Введем целочисленные переменные *tij*, означающие время начала выполне- ния работы *j* на машине *i*. Введем *mnn* булевых переменных, чтобы задать условия непересечения работ при выполнении на одной машине:

*xijk* =

на машине *i, j < k,*

 0 − в противном случае.

 1*,* если работа *j* предшествует работе *k*

Возникают взаимоисключающие друг друга условия. Если работа *j* пред- шествует работе *k* на машине *i*, то время начала работы *k* должно наступить не раньше времени завершения работы *j*:

*tik* ≥ *tij* + *pij,* если *xijk* = 1

и

*tij* ≥ *tik* + *pik,* если *xijk* = 0*.*

Используя рассуждения изложенные в этом разделе, эти условия можно реа- лизовать с помощью следующих неравенств:

*tik* ≥ *tij* + *pij* − *W* (1 − *xijk*)*,*

*tij* ≥ *tik* + *pik* − *W xijk ,*

где *W* — большое положительное число.

Каждая работа состоит из операций, выполняемых на разных машинах, и (*r* + 1)-я операция работы *j* не может начаться, пока не будет завершена предыдущая *r*-я операция, значит,

*tj*(*r*+1)*j* ≥ *tj*(*r*)*j* + *pj*(*r*)*j .*

Итак, математическая модель выглядит следующим образом. Целевая функция

— минимальная сумма времен завершения всех работ:

при ограничениях:

*n*

min (*tj*(*m*)*j* + *pj*(*m*)*j*) (2.54)

∑

*j*=1

*tj*(*r*+1)*j* ≥ *tj*(*r*)*j* + *pj*(*r*)*j, r* = 1*, . . . ,* (*m* − 1)*, j* = 1*, . . . , n,*

29

*tik* ≥ *tij* + *pij* − *W* (1 − *xijk*)*, i* = 1*, . . . , m*; *j, k* = 1*, . . . , n,*

*tij* ≥ *tik* + *pik* − *W xijk, i* = 1*, . . . , m*; *j, k* = 1*, . . . , n,*

*tij* ≥ 0*, i* = 1*, . . . , m*; *j* = 1*, . . . , n,*

*xijk* ∈ {0*,* 1}*, i* = 1*, . . . , m*; *j, k* = 1*, . . . , n.*

Заметим, что в целевой функции (2.54) величина ∑*n pj*(*m*)*j* является постоян-

*j*=1

ной и не зависит от порядка выполнения работ. Следовательно, оптимальное ре-

шение не изменится, если целевую функцию (2.54) заменить на min ∑*n*

*j*=1

*tj*(*m*)*j .*

## Линеаризация в математических моделях

## Линеаризация произведения переменных

Если в задаче встречается квадратичное выражение вида *xixj*, где *xi, xj*

∈

0*,* 1 , то для него можно предложить эквивалентную линейную переформули- ровку. Для этого введем новую булеву переменную *yij*, такую, что *yij* = *xixj*, т. е.

{ }

*yij* = 1 тогда и только тогда, когда *xi* = 1 и *xj* = 1*,*

другими словами,

если *yij* = 1*,* то *xi* = 1*,*

если *yij* = 1*,* то *xj* = 1

и

если *xi* = 1 и *xj* = 1*,* то *yij* = 1*.*

Применяя первое и второе правила к трем импликациям, получаем три следу- ющих неравенства:

1 − *xi* ≤ 1 − *yij,*

1 − *xj* ≤ 1 − *yij,*

1 − *yij* ≤ 1 − *xi* + 1 − *xj.*

Или в более упрощенной форме:

|  |  |
| --- | --- |
| *yij* ≤ *xi,* | (2.55) |
| *yij* ≤ *xj,* | (2.56) |
| *xi* + *xj* − *yij* ≤ 1*.* | (2.57) |
| *yij, xi, xj* ∈ {0*,* 1} | (2.58) |

Интересное и полезное наблюдение с точки зрения линейного программирова- ния заключается в том, что даже если отказаться от булевости переменных *yij* и перейти к непрерывным значениям из отрезка [0*,* 1], то неравенства (2.55)– (2.58) будут выполняться, только когда переменные *yij* равны 0 или 1.

30

*Пример. Задача о клике*

Задан неориентированный граф *tt* с множеством вершин *V* и множеством ребер *E*. Задача заключается в том, чтобы найти максимальный (по количеству вершин) полный подграф, т. е. клику. Напомним, что простой граф без петель и кратных ребер называется *полным*, если любая пара вершин соединена ребром.

Введем следующие булевы переменные:

{*x* =*v*

1*,* если вершина *v* входит в подграф,

0 − в противном случае.

Тогда целевая функция найти клику максимальной мощности:

∑

max *xv.* (2.59)

*v*∈*V*

Подграф является кликой тогда и только тогда, когда он не содержит пару вершин, между которыми нет ребра в исходном графе, т. е.

если *xw* = 1*,* то *xv* = 0*,* для (*v, w*) ∈*/ E.*

С учетом описанных выше правил, эта импликация моделируется неравен- ством:

*xv* ≤ 1 − *xw* для (*v, w*) ∈*/ E.* (2.60) Добавив ограничения на значения переменных:

*xv* ∈ {0*,* 1}*, v* ∈ *V,* (2.61)

получаем постановку задачи о клике в виде математической модели (2.59)– (2.61) целочисленного линейного программирования.

Задача о клике часто возникает в приложениях, связанных с телекомму- никационными, транспортными и другими сетями. В них могут быть заданы веса ребер или вершин и ищется максимальная по весу клика. Кроме того, в таких задачах, как правило, присутствует много других ограничений, среди которых — требования полноты подграфа. Целевая функциях в таких задачах может зависеть от суммарного веса ребер, входящих в клику. И тогда кроме вершин, образующих максимальную клику, нужно знать еще и ребра, входя- щие в полный подграф. В этой ситуации переменных, соответствующих только вершинам, недостаточно, и вводятся дополнительные булевые переменные:

 1*,* если ребро между вершинами *v* и *w*

*yvw* =

входит в подграф,

 0 − в противном случае.

Ребро (*v, w*) из множества *E* входит в подграф тогда и только тогда, когда обе вершины *v* и *w* входят в подграф. Следовательно, *yvw* = *xvxw*. Произведе- ние можно линеаризовать, переписав в виде линейной системы неравенств. Из импликации

*yvw* = 1 тогда и только тогда, когда *xv* = 1 и *xw* = 1 для (*v, w*) ∈ *E,*

31

пользуясь правилами, получаем:

1 − *yvw* ≤ (1 − *xv*) + (1 − *xw*)*,* (*v, w*) ∈ *E,* (2.62)

*yvw* ≤ *xv,* (*v, w*) ∈ *E,* (2.63)

*yvw* ≤ *xw,* (*v, w*) ∈ *E.* (2.64)

Итак, неравенства (2.60)–(2.64) гарантируют, что выбранный подграф будет кликой.

Заметим, что можно предложить другой вариант моделирования полноты подграфа. Рассмотрим произвольные четыре вершины. Если одна пара вершин в графе соединена ребром и другая пара вершин соединена ребром, а между какими-либо вершинами из этих пар ребра в графе нет, то в подграф должна входить только одна пара из этих вершин. Формально для любых *s*, *t*, *u*, *v* ∈ *V* , таких, что (*s, t*) ∈*/ E*, (*s, u*) ∈ *E* и (*t, v*) ∈ *E*,

если *ysu* = 1*,* то *ytv* = 0*,*

в аналитической форме

*ytv* ≤ 1 − *ysu.* (2.65)

Эту формулировку можно получить из неравенств (2.60)–(2.64). Просумми- ровав пару неравенств типа (2.63) для (*s, u*) *E* и (*t, v*) *E*, получим *ysu* + *ytv xs* + *xt.* Поскольку (*s, t*) */ E*, то из (2.60) получаем *xs* + *xt* 1, следовательно, неравенство (2.65) выполняется.

≤ ∈ ≤

∈ ∈

Последний вариант модели содержит меньше переменных, их столько, сколь- ко ребер в исходном графе, но существенно больше ограничений по сравнению с (2.60)–(2.64).

## Линеаризация заменой переменных

*Пример. Задача размещения с распределенными закупками*

Предпринимателю известно конечное множество *I* возможных мест для от- крытия *p* торговых центров и конечное множество потребителей *J* . Под потре- бителями можно понимать не индивидуальное лицо, а «типичного» представи- теля, который характеризует поведение некоторой группы людей. Потребители выбирают открытый торговый центр и расходуют деньги пропорционально сво- им предпочтениям. Предпочтение торгового центра *i* потребителем *j* измеря- ется величиной *uij*, *uij* 1 для всех *i I*, *j J* . Бюджет каждого потребителя ограничен величиной *Bj*. Предприниматель оценивает свою прибыль от каж- дой потраченной потребителем *j* денежной единицы в магазине *i* величиной *cij*. Задача предпринимателя — открыть *p* торговых центров так, чтобы получить максимальную прибыль.

≥ ∈ ∈

Введем булевые переменные:

*xi* = {

1*,* если в месте *i* открывается торговый центр,

0 − в противном случае

32

и неотрицательные вещественные переменные *yij*, которые означают сумму, потраченную клиентом *j* в торговом центре *i*.

Целевая функция — максимальная суммарная прибыль:

∑ ∑

max *cijyij* (2.66)

*i*∈*I j*∈*J*

при ограничениях на число открываемых предприятий:

∑

*xi* = *p* (2.67)

*i*∈*I*

и при условиях, что каждый потребитель тратит свой бюджет во всех открытых торговых центрах пропорционально предпочтениям:

*uijxi*

(2.68)

*yij* = *Bj* ∑

*k*∈*I*

*ukj xk*

*, i* ∈ *I, j* ∈ *J,*

*xi* ∈ {0*,* 1}*, yij* ≥ 0*, i* ∈ *I, j* ∈ *J.* (2.69)

Недостатком этой модели является нелинейная связь между переменными

*xi* и *yij* в ограничениях (2.68).

*Линеаризация модели*

Сделаем замену переменных. Введем новые неотрицательные вещественные переменные *zj*, *j* ∈ *J* , такие, что

*Bj*

(2.70)

*zj* = ∑

*k*∈*I*

*.*

*ukj xk*

Тогда целевая функция и ограничение на число открываемых предприятий остаются без изменений. Условия, что клиенты могут посещать торговый центр, только если он открыт, записывается следующим образом:

*yij* ≤ *Bjxi, i* ∈ *I, j* ∈ *J.* (2.71) Потребители тратят весь свой бюджет:

∑

*yij* = *Bj, j* ∈ *J.* (2.72)

*i*∈*I*

Следующая группа ограничений устанавливает связь между переменными *yij*

и *zj* и определяет значения переменных *zj* в соответствии с равенством (2.70):

*yij* ≤ *uijzj, i* ∈ *I, j* ∈ *J,* (2.73)

*uijzj* ≤ *yij* + *Bj*(1 − *xi*)*, i* ∈ *I, j* ∈ *J,* (2.74)

33

ограничения на значения переменных:

*yij* ≥ 0*, zj* ≥ 0*, xi* ∈ {0*,* 1}*, i* ∈ *I, j* ∈ *J.* (2.75)

Группы ограничений (2.73), (2.74) означают, что если *i*-й торговый центр открыт, т. е. *xi* = 1, то значение *yij* будет совпадать со значением *uijzj*. Если же *i*-й торговый центр закрыт, *xi* = 0, то благодаря наличию в ограничении (2.74) достаточно большого числа *Bj* неравенство остается верным.

*Пример. Задача о ценообразовании*

Фирма-производитель предлагает потребителям однородную продукцию в нескольких филиалах по разным ценам. Каждый потребитель выбирает наи- более выгодный филиал, учитывая транспортные затраты до него, свой бюджет и стоимость продукции в филиале. Если сумма транспортных затрат и стоимо- сти продукции превышает бюджет клиента, то клиент не покупает продукцию. Если же бюджета хватает на покупку и несколько филиалов дают минималь- ные затраты для клиента, то клиент выбирает филиал с минимальными транс- портными затратами. Задача фирмы — назначить такую стоимость продукции в каждом филиале, чтобы получить максимальный суммарный доход от своих филиалов.

Обозначим через *I* множество филиалов, через *J* множество потребителей; *bj* — бюджет *j*-го потребителя; *cij* — транспортные затраты от *i*-го филиала до *j*-го потребителя.

Введем неотрицательные переменные *pi* — стоимость продукции в *i*-м фи- лиале и булевые переменные:

{*x* =*ij*

1*,* если *j*-й потребитель выбрал *i*-й филиал,

0 − в противном случае.

Целевая функция задачи — максимальный суммарный доход фирмы:

max ∑ *pi* ∑ *xij.* (2.76)

*i*∈*I*

*j*∈*J*

*i*∈*I*

*j*∈*J*

Каждый потребитель выбирает не более одного филиала:

*xij* ≤ 1*, j* ∈ *J.* (2.77)

∑

*i*∈*I*

Затраты каждого потребителя не должны превышать его бюджет:

∑

(*bj* − *cij* − *pi*)*xij* ≥ 0*, j* ∈ *J.* (2.78)

*i*∈*I*

Из всех филиалов каждый потребитель выбирает тот филиал, который до- ставляет минимальные суммарные затраты, включающие в себя транспортные расходы и расходы на покупку продукции:

∑

(*cij* + *pi*)*xij* ≤ *ckj* + *pk, k* ∈ *I, j* ∈ *J.* (2.79)

*i*∈*I*

34

Ограничения на принимаемые значения переменных:

*pi* ≥ 0*, xij* ∈ {0*,* 1}*, i* ∈ *I, j* ∈ *J.* (2.80)

Недостатком этой модели является нелинейность целевой функции (2.76) и ограничений (2.78), (2.79).

*Линеаризация модели для задачи о ценообразовании*

Обозначим через *pi* максимально возможную цену в *i*-м филиале: *pi* = max*j*∈*J* (*bj cij*). Введем неотрицательные вещественные переменные *zij*, от- вечающие за доход, который получает производитель от *i*-го филиала и *j*-го

−

потребителя в нем: *zij* = *pixij*. Используя данные обозначения, получим следу- ющую модель:

∑ ∑

при ограничениях:

max *zij* (2.81)

*i*∈*I j*∈*J*

∑(*bj* − *cij*)*xij* − ∑ *zij* ≥ 0*, j* ∈ *J,* (2.82)

*i*∈*I*

*i*∈*I*

*i*∈*I*

*i*∈*I*

*ckj* + *pk* − ∑ *cijxij* − ∑ *zij* ≥ 0*, k* ∈ *I, j* ∈ *J,* (2.83)

*i*∈*I*

*i*∈*I*

*i*∈*I*

*i*∈*I*

(1 − *xij*)*pi* − *zij* + *pi* ≥ 0*, i* ∈ *I, j* ∈ *J,* (2.84)

(1 − *xij*)*pi* + *zij* − *pi* ≥ 0*, i* ∈ *I, j* ∈ *J,* (2.85) *zij* ≤ *pixij, i* ∈ *I, j* ∈ *J,* (2.86) *xij* ≤ 1*, j* ∈ *J,* (2.87)

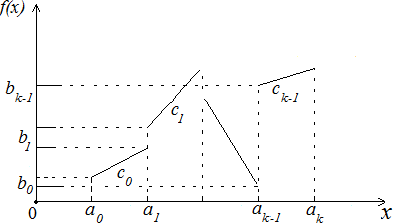
∑

*i*∈*I*

*pi* ≥ 0*, xij* ∈ {0*,* 1}*, zij* ≥ 0*, i* ∈ *I, j* ∈ *J.* (2.88) Ограничения (2.84)–(2.86) гарантируют, что доход фирмы *zij* от обслужи-

вания *j*-го потребителя в *i*-м филиале равен *pi*, если *j*-й потребитель выбрал *i*-й филиал, т. е. *xij* = 1, и равен нулю из условий (2.86) при *xij* = 0. По сути, ограничения (2.84)–(2.86) заменяют равенство *zij* = *pixij*, что и приводит к линейной модели.

35



*Рис. 5.* Кусочно-линейная функция

## Линеаризация кусочно-линейной функции

Пусть на отрезке [*a*0*, ak*] задана кусочно-линейная функция (рис. 5):



0*,* если *x* = *a*0*,*

*b*0 + *c*0(*x* − *a*0)*,* если *a*0 *< x* ≤ *a*1*,*

*b*





0

*f* (*x*) =

.



0

1

0

1

1

1

если *a*1 *< x* ≤ *a*2*,*

+ *c* (*a*

*i*=1

− *a* ) + *b*

+ *c* (*x* − *a* )*,*

∑*k*−1[*bi*−1 + *ci*−1(*ai* − *ai*−1)] + *bk*−1 + *ck*−1(*x* − *ak*−1)*,*

 если *a < x* ≤ *a ,k*−1 *k*



Коэффициенты *bi*, *ci*, *i* = 0*, . . . ,* (*k* 1) — действительные числа. Требуется смоделировать поиск минимума функции *f* (*x*), при *a*0 *x ak*.

≤ ≤

−

Введем булевые переменные:

{*y* =*i*

1*,* если *x > ai*

0 − в противном случае

и неотрицательные переменные *zi* — столько от числа *x* лежит в интервале

[*ai, ai*+1], т. е. *zi* = *x ai*.

−

MIP для поиска минимума *f* (*x*) выглядит следующим образом:

*k*−1

∑

min (*biyi* + *cizi*)

*i*=0

при ограничениях:

(*a*1 − *a*0)*y*1 ≤ *z*0 ≤ (*a*1 − *a*0)*y*0*,*

36

(*a*2 − *a*1)*y*2 ≤ *z*1 ≤ (*a*2 − *a*1)*y*1*,*

.

0 ≤ *zk*−1 ≤ (*ak* − *ak*−1)*yk*−1*,*

*k*−1

∑

*x* = *zi* + *a*0*,*

*i*=0

*x* ≥ 0*, yi* ∈ {0*,* 1}*, zi* ≥ 0*, i* = 0*, . . . ,* (*k* − 1)*.*

Проверим, что если (*y, z, x*) — допустимое решение, удовлетворяющее ограни-

чениям в этой модели, то *f* (*x*) = ∑*k*−1(*biy* + *cizi*).

≤ ≤ −−

*i*=0

*i*

Рассмотрим *y*. Предположим, что *yr* 1 = 1, *yr* = 0, 1 *r* (*k* 1)*.* Т.к.

*yr* = 0, то

{

т.к. *yr*−1 = 1, то

*yi* = 0 при *i* ≥ *r, zi* = 0 при *i* ≥ *r,*

*yi* = 1 при *i* (*r* 1)*,*



≤ −

*z* = *a a* при *i* (*r* 1)*,*

− ≤ −*i*−1 *i i*−1

 0 ≤ *zi*−1 ≤ *ai* − *ai*−1 при *i* = *r.*

*i*

Следовательно, ∑*k*−1(*biy* + *cizi*)= ∑*r*−1 *bi* + ∑*r*−1 *ci*−1(*ai* − *ai*−1) +*cr*−1*zr*−1=

*i*=0

*i*=0

*i*=1

*i*=0

*i*=1

∑*r*−1 *bi*+ ∑*r*−1 *ci*−1(*ai* − *ai*−1)+*cr*−1(*x* − *ar*−1)*.* Последнее равенство следует из

того, что по построению *x* = *ar*−1 + *zr*−1 при 0 ≤ *zr*−1 ≤ (*ar* − *ar*−1), следова-

тельно *f* (*x*) = ∑*k*−1(*biy* + *cizi*).

Заметим, что необходимость в моделировании кусочно-линейной функции линейными ограничениями может возникнуть, когда целевая функция задачи более сложная и представляет собой линейную функцию, включающую в себя кусочно-линейную.

*i*=0

*i*

## 2.7. Симметрия в математических моделях

Один из недостатков, которым могут обладать математические модели, яв- ляется симметрия в допустимых решениях. Этот недостаток приводит к тому, что точные методы, типа ветвей и границ, будут тратить время на просмотр и проверку эквивалентных решений. Рассмотрим один из вариантов симметрии и как от нее избавиться на примере задачи кластеризации [26].

Пусть требуется разбить конечное множество объектов *I* на *p* групп. Каж- дый объект может попасть только в одну группу. В разд. 2.3 уже рассматри- вались задачи кластеризации, но переменные, которые использовались там, в данном случае будут неудобными, поскольку сейчас необходимо отслеживать количество групп. Введем другие булевы переменные:

*xik* = {

1*,* если объект *i* попадает в группу *k,*

0 − в противном случае,

37

где *i I*, *k* = 1*, . . . , p.* Каждый объект должен попасть только в одну группу, значит,

∈

*p*

∑

*xik* = 1*.* (2.89)

*k*=1

Сложность возникает в том, что группы можно пронумеровать по-разному, и в результате будут получаться формально разные решения, но по содержа- щимся в группах объектам это будет одно и то же решение. Например, для *I* = {*a, b, c, d*}, *p* = 3 и разбиения {*a*}, {*b, d*}, {*c*} следующие решения эквива-

лентны:

*Таблица 4*

Эквивалентные решения

|  |  |  |  |
| --- | --- | --- | --- |
| Группа 1 | Группа 2 | Группа | 3 |
| *{b, d}* | *{a}* | *{c}* | |
| *{b, d}* | *{c}* | *{a}* | |
| *{a}* | *{b, d}* | *{c}* | |
| *{a}* | *{c}* | *{b, d}* | |
| *{c}* | *{b, d}* | *{a}* | |
|  |  |  | |

Таким образом, каждому разбиению множества объектов соответствует (*p*!) эквивалентных решений, отличающихся друг от друга нумерацией групп. Воз- никает необходимость избавиться от перестановочной симметрии и лишних, по сути, решений, оставив только одно из них. Одним из способов может быть установление лексикографического порядка на множестве решений. Для этого упорядочим объекты множества *I* от 1 до *I* .

| |

Среди множества эквивалентных решений выберем одно следующим обра- зом. В каждой группе найдем объект с наименьшим номером. Упорядочим группы по возрастанию этих номеров и пронумеруем группы от 1 до *p* соглас- но полученному порядку. Построенное таким образом решение будем называть *лексикографически минимальным решением* среди эквивалентных ему реше- ний.

Например, разобьем множество 1*, . . . ,* 10 на группы 4*,* 6*,* 8*,* 9 , 1*,* 2*,* 7 , 3*,* 5*,* 10 *.* Согласно установленному выше лексикографическому порядку груп- па 1*,* 2*,* 7 будет первой, 3*,* 5*,* 10 — второй, 4*,* 6*,* 8*,* 9 — третьей.

{ } { } { }

{ }

{ } { } { }

Разберем теперь, как шаг за шагом, пользуясь описанными уже правилами 1 и 2, построить линейные ограничения, реализующие предложенный подход.

*Назначения в группу 1.* Объект 1 должен лежать в группе 1, т. е. *x*11 = 1*.*

Заметим, что остальные переменные *x*1*k*, при *k >* 1, равны нулю.

*Назначения в группу 2.* Если объект 2 не лежит в группе 1, то он должен

лежать в группе 2, т. е.

если *x*21 = 0*,* то *x*22 = 1*.*

38

Получаем неравенство:

(1 − *x*22) ≤ *x*21*.*

Если объект 2 лежит в группе 1 и объект 3 не лежит в группе 1, то он должен лежать в группе 2, т. е.

если *x*21 = 1 и *x*31 = 0*,* то *x*32 = 1*.*

Получаем неравенство:

(1 − *x*32) ≤ (1 − *x*21) + *x*31*.*

Далее если объекты 2*, . . . ,* (*j* 1) лежат в группе 1 и объект *j* не лежит в группе 1, тогда объект *j* должен быть в группе 2, т. е.

−

если *xi*1 = 1 для всех *i* = 2*, . . . ,* (*j* − 1) и *xj*1 = 0*,* то *xj*2 = 1*.*

Получаем неравенство:

(1 − *xj*2) ≤

(*j*−1)

(1 − *xi*1) + *xj*1*.*

∑

*i*=2

*Назначения в группы c 3 по* (*p* 1)*.* В каждую группу *k* (*k* = 3*, . . . , p*

− −

1. должен попасть объект с наименьшими номером, который еще не попал в предыдущую группу с номерами от 1 до (*k* 1). Значит, для любого объекта с номером *j*, если все предыдущие объекты с номерами от 2 до (*j* 1) лежат в группе *l*, *l < k*, а *j* не лежит в *l*, то *j* должен быть в группе *k*.

−

−

Вспомним, что каждый объект принадлежит только одной группе, значит *i*

может лежать в группе с меньшим номером *l* (меньшим, чем *k*) тогда и только

∑ −тогда, когда *x* = 1*.il*

*k* 1

*l*=1

Таким образом, чтобы еще не размещенный ни в одной группе объект *j*

определял следующий номер группы c 3 по (*p* 1), должно быть выполнено условие: *если* ∑(*k*−1) *xil* = 1 *и* ∑(*k*−1) *xjl* = 0 *для всех i* = 2*, . . . , j* − 1*, то*

*xjk* = 1*.*

−

*l*=1

*l*=1

Следуя правилам 1 и 2, смоделируем эту импликацию в виде следующих неравенств:

*j*−1

*k*−1

*k*−1

(1 − *xjk*) ≤ ∑(1 − ∑ *xil*) + ∑ *xjl, j* = *k, . . . , n, k* = 3*, . . . ,* (*p* − 1)

*i*=2

*l*=1

*l*=1

*i*=2

*l*=1

*l*=1

или после приведения подобных:

*j*−1 *k*−1 *k*

∑ ∑ *xil* − ∑ *xjl* ≤ *j* − 3*, j* = *k, . . . , n, k* = 3*, . . . ,* (*p* − 1)*.*

*i*=2 *l*=1

*l*=1

39

Заметим, что переменные *xik* с индексами *i < k* не нужны, так как соглас- но лексикографическому порядку никакой объект не может быть размещен в группе с большим, чем у самого объекта, номером.

*Назначения в группу с номером p.* Нет необходимости в том, чтобы выпи- сывать условия для последней группы с номером *p*, поскольку все объекты, не размещенные в группы с меньшими номерами, автоматически будут назначены в последнюю.

40

# Примеры математических моделей целочисленного линейного программирования

В задачах *комбинаторной* оптимизации оптимальное решение требуется вы- брать из конечного множества возможных решений. Этот раздел содержит ряд примеров задач комбинаторной оптимизации, для которых можно запи- сать математические модели с целочисленными неотрицательными перемен- ными и линейными связями между ними. В частности, для одного из основных представителей этого класса, задачи коммивояжера, будет описано несколько разных моделей, отличающихся количеством переменных и ограничений. Будет представлена модель целочисленного линейного программирования для задачи о двухстадийном гильотинном раскрое материала. Также будут рассмотрены примеры для хорошо известных в дискретной оптимизации задач о потоке и планировании производства интересные с точки зрения приемов моделирова- ния.

## Задача о потоке минимальной стоимости

В общем виде задача формулируется следующим образом. Задана сеть — ориентированный граф с множеством вершин *V,* в которых могут размещаться предприятия (поставщики, производители и пр.) некоторой продукции, множе- ством дуг *A* и двумя специальными вершинами: источником и стоком. Дуга (*i, j*) означает, что из вершины *i* в вершину *j* может осуществляться поставка некоторой продукции. Каждая дуга имеет неотрицательный вес *uij*, соответ- ствующий пропускной способности этой дуги. Известны *cij* — удельные стои- мости перевозки продукции вдоль дуги (*i, j*). Каждая вершина *i* имеет вес *bi*. В зависимости от знака величины *bi* вес вершины может означать:

* + спрос на продукцию в этой вершине (или сколько продукции должно быть доставлено в вершину *i*), если *bi <* 0;
  + предложение продукции в этой вершине (или сколько продукции можно вывести из вершины *i*), если *bi >* 0;
  + транзитная вершина (продукция не остается и не забирается из вершины

*i*), если *bi* = 0.

∑ ∈

Предполагается, что *i V bi* = 0*.*

Задача заключается в том, чтобы доставить в каждую вершину нужное

количество продукции из вершин, в которых она находится, с минимальными затратами, не нарушая пропускных возможностей дуг.

Введем неотрицательные переменные *xij* — величина потока, пропускаемого по дуге (*i, j*).

Целевая функция — минимальные суммарные затраты на перевозки внутри сети:

∑

min

(*i,j*)∈*A*

*cijxij* (3.1)

41

при ограничениях на пропускные способности для каждой дуги:

*xij* ≤ *uij,* (*i, j*) ∈ *A,* (3.2)

для каждой вершины *i* должно выполняться *условие баланса* между потоком, входящим в эту вершину, потоком, выходящим из нее, и количеством продук- ции, доставляемой или вывозимой из этой вершины:

∑ *xji* − ∑ *xij* = *bi, i* ∈ *V,* (3.3)

*j*∈*V*

*j*∈*V*

*j*∈*V*

*j*∈*V*

и ограничениях на значения переменных:

*xij* ≥ 0*,* (*i, j*) ∈ *A.* (3.4)

Полученная модель линейного программирования (3.1)–(3.4) соответствует *за- даче о потоке минимальной стоимости*.

В некоторых задачах за осуществление перевозок, кроме удельных затрат, взимаются затраты, которые не зависят от объемов перевозок. Пусть *hij* — пла- та за использование дуги (*i, j*). В этом случае нужны дополнительные булевы переменные:

{*y* =*ij*

1*,* если по дуге (*i, j*) выполняются перевозки,

0 − в противном случае.

Целевая функция изменится, в ней появятся дополнительные слагаемые:

min ∑ (*cijxij* + *hijyij*)*.* (3.5)

(*i,j*)∈*A*

Переменные *xij* и *yij* связаны следующим образом:

если *yij* = 0*,* то *xij* = 0*,* причем 0 ≤ *xij* ≤ *uij.*

Согласно следствию из первого правила (разд. 2.2) эта зависимость моделиру- ется следующим образом:

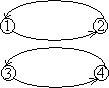
*xij* ≤ *uijyij,* (*i, j*) ∈ *A.* (3.6)

Модель (3.3)–(3.6) с *yij* 0*,* 1 , для всех (*i, j*) *A*, является моделью сме- шанного целочисленного программирования и соответствует *задаче о потоке с фиксированными затратами* [30].

∈ { } ∈

Подобные задачи о потоках возникают при моделировании систем подачи воды, отопления. Этот класс задач обладает важным свойством: если все веса *bi* и пропускные способности *uij* являются целыми числами, то оптимальное решение будет целочисленным. Это свойство задачи будет важным при обсуж- дении вопроса о разрыве целочисленности в разд. 4.

42



*Рис. 6.* Несколько циклов

## Задача коммивояжера

Дано *n* городов. Известно *cij* — время перемещения из города *i* в город *j*. Коммивояжер выезжает из какого-то города, объезжает все города и возвра- щается в исходный. Необходимо найти такую последовательность посещения городов, при которой суммарное время перемещения было бы минимальным.

Введем *n*2 булевых переменных *xij*, *i* = 1*, . . . , n*, *j* = 1*, . . . , n*:

*xij* =

в город *j,*

 0 − в противном случае.

 1*,* если коммивояжер едет из города *i*

Тогда суммарное время передвижения составляет

*n n*

∑ ∑

*cij xij .*

*i*=1 *j*=1

Однократный выезд из города *i* задается условием

*n*

∑

*xij* = 1*,* (3.7)

*j*=1

однократный въезд в город *i* задается условием

*n*

∑

*xji* = 1*.* (3.8)

*j*=1

Однако этих условий недостаточно, так как может возникать несколько циклов. Например, при *n* = 4 решение со значениями переменных *x*12 = *x*21 = *x*34 = *x*43 = 1 удовлетворяет ограничениям (3.7), (3.8), но не является решением задачи, так как образует два цикла, как показано на рис. 6.

В 1954 г. Данциг предложил следующий способ исключения такой ситуации. Пусть имеется два цикла. Чтобы получить из них обход всех городов, нужно,

43

чтобы вместо одного ребра между парой вершин в цикле была пара ребер, соединяющая вершины, не попавшие в этот цикл. Значит, хотя бы для одной пары городов *i* и *j*, лежащих в разных циклах, переменная *xij* должна быть равна 1.

Обозначим через *I* множество всех городов. Пусть *J* — некоторое подмно- жество множества *I*, тогда условия, запрещающие всевозможные подциклы, выглядят следующим образом:

∑ ∑ *xij* ≥ 1*,* ∀*J* ⊂ *I*

или

*i*∈*J j*∈*I*\*J*

*xij* ≤ |*J* | − 1*,* ∀*J* ⊂ *I.*

∑ ∑

*i*∈*J j*∈*J*

В результате получаем следующую математическую модель в терминах це- лочисленного линейного программирования:

*n n*

∑ ∑

min *cijxij* (3.9)

*i*=1 *j*=1

*n*

∑

*xij* = 1*, i* ∈ *I,* (3.10)

*j*=1 *n*

∑

*xji* = 1*, i* ∈ *I,* (3.11)

*j*=1

∑ ∑ *xij* ≥ 1*, J* ⊂ *I,* 2 ≤ |*J* | ≤ |*I*| − 2*,* (3.12)

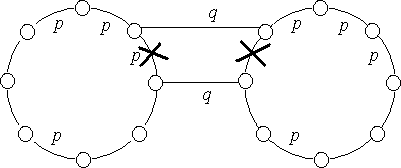
*i*∈*J j*∈*I*\*J*

*xij* ∈ {0*,* 1}*, i, j* ∈ *I.* (3.13)

Заметим, что задача о назначениях, рассмотренная в разд. 1, тесно свя- зана с задачей коммивояжера. Максимальное количество подциклов в задаче о назначениях равно *n*, когда каждая машина *i* назначается на выполнение работы *i*. Минимальное количество циклов равно 1, и тогда такое назначение соответствует решению задачи о коммивояжере. Таким образом, множество до- пустимых решений в задаче о назначениях шире, чем множество допустимых решений в задаче о коммивояжере, поэтому оптимальное решение задачи о назначениях дает нижнюю оценку для оптимального решения задачи о комми- вояжере. Покажем, что такая оценка может сколько угодно сильно отличаться от оптимального решения задачи коммивояжера.

Предположим, что оптимальное решение задачи о назначениях состоит из двух подциклов. Пусть элементы *cij*, участвующие в подциклах, равны *p*, а остальные элементы равны *q*, *q > p* (рис. 7).

44



*Рис. 7.* Ликвидация подциклов в задаче коммивояжера

Чтобы получить оптимальное решение задачи о коммивояжере, т. е. один цикл, необходимо «разорвать» первый и второй подциклы, как показано на рис. 7, т. е. удалить два ребра длиной 2*p* и добавить два ребра длиной 2*q*. Если в первом подцикле *k* ребер, а во втором — *s*, то значение целевой функции в задаче о назначениях на оптимальном решении равно (*k* + *s*)*p*. Для оптималь- ного решения задачи о коммивояжере получаем значение целевой функции, равное (*n* 2)*p* + 2*q*. Тогда разница между оптимальными значениями состав- ляет 2(*q p*)*.* Следовательно, при фиксированном значении *p* величина *q* может быть выбрана так, что разность 2(*q p*) может быть сколько угодно большой. Недостатком приведенной модели является то, что число ограничений экс-

−

−

−

поненциально, их (*n* + *n* + (2*n*−1 *n* 1)). В работе Таккера в 1960 г. была предложена постановка с полиномиальным числом ограничений.

− −

Рассмотрим *n* целочисленных переменных *ui*, соответствующих номеру ша- га, на котором посетили город *i*, тогда ограничения

*ui* − *uj* + *nxij* ≤ *n* − 1*, i, j* = 2*, . . . , n, i* ̸= *j* (3.14)

запрещают подциклы. Проверим, что эти условия действительно гарантируют наличие ровно одного цикла, проходящего через все города.

Пусть имеется 2 цикла. Один из них не проходит через первый город, обозна- чим его (*i*2*, . . . , ik, i*2). Выпишем для каждой пары городов, идущих по порядку в этом цикле, условия (3.14):

*ui*2 − *ui*3 + *n* ≤ *n* − 1*, ui*3 − *ui*4 + *n* ≤ *n* − 1*,*

. 45

*uik* − *ui*2 + *n* ≤ *n* − 1*.*

Сложив эти неравенства, получим *kn* (*n* 1)*k*, что невозможно при *k* = 0.

≤ − ̸

Проверим, что цикл, проходящий через все города, удовлетворяет условиям (3.14), т. е. можно подобрать соответствующие значения переменных *ui*. Пусть *ui* = *k*, если город *i* посетили на шаге *k*, тогда неравенство *ui uj n* 1 верно при *xij* = 0. Если *xij* = 1, то *ui* = *k*, а *uj* = (*k* + 1), тогда *k* (*k* + 1) + *n n* 1, *n* 1 *n* 1.

− ≤ −

− ≤ −

− ≤ −

Итак, (3.9)—(3.11), (3.13), (3.14), *ui* 0*, i I* — математическая модель задачи коммивояжера с полиномиальным числом ограничений.

≥ ∈

Несмотря на то, что в этой модели существенно меньше переменных, она обладает недостатком, о котором речь пойдет в разд. 4.

Рассмотрим еще одну постановку задачи коммивояжера с полиномиальным числом переменных и ограничений, основанную на идеях моделирования зада- чи о потоке.

Оставим прежние булевы переменные *xij* и введем полиномиальное количе- ство новых переменных: *yij* — поток некоторой продукции между городами *i* и *j*, тогда ограничения (3.10), (3.11) и следующий набор ограничений представ- ляют все гамильтоновы циклы в задаче о коммивояжере:

*yij* − (*n* − 1)*xij* ≤ 0*, i, j* ∈ *I,* (3.15)

∑

*y*1*j* = *n* − 1*,* (3.16)

*j*∈*I*

∑ *yij* − ∑ *yki* = −1*,* для всех *i* ∈ *I, i* ̸= 1*,* (3.17)

*j*∈*I k*∈*I*

*yij* ≥ 0*, i, j* ∈ *I,* (3.18)

*xij* ∈ {0*,* 1}*, i, j* ∈ *I.* (3.19)

Ограничения (3.15) гарантируют перевозку продукции между городами *i* и *j*, только если *xij* = 1. Ограничение (3.16) говорит о том, что из первого города нужно развести (*n* 1) единиц товара по остальным городам, оставляя, соглас- но (3.17), в каждом городе ровно одну единицу товара. Чтобы развести всю продукцию, сеть должна быть связанной, таким образом, подциклов в обходе возникать не будет. Итак, в этой модели (*n*2 + 3*n*) ограничений и 2*n*2 перемен- ных.

−

## Задача о покрытии

Задано *n* возможных мест расположения пожарных станций. Величины *cj*, *j* = 1*, . . . , n* задают стоимость размещения пожарной станции в месте *j*. Из- вестно *m* объектов, на которые должны выезжать пожарные бригады. Причем пожарной бригаде имеет смысл выезжать на объект, если он находится на рас- стоянии не более *D* км, поэтому все объекты разбиты на группы объектов *Mj*, которые можно достичь из пожарной станции *j*.

46

Нужно определить, в каких местах разместить пожарные станции, чтобы все объекты были достижимы из действующих пожарных станций, а суммарные затраты на их размещение были бы минимальными.

Построим вспомогательную матрицу (*aij*)*m*×*n*, чтобы задать разбиение объ- ектов на группы:

{*a* =*ij*

1*,* если объект *i* попадает в группу *Mj,*

0 − в противном случае. Введем *n* булевых переменных *xj*, *j* = 1*, . . . , n*:

{*x* =*j*

1*,* если в пункте *j* размещается пожарная станция,

0 − в противном случае.

Целевая функция задачи — минимальные суммарные затраты на открытие станций:

∑

*n*

min *cjxj* (3.20)

*j*=1

при ограничениях, что каждый объект должен быть достижим хотя бы из од- ной пожарной станции:

*n*

∑

*aijxj* ≥ 1*, i* = 1*, . . . , m* (3.21)

*j*=1

и ограничения на значения переменных:

*xj* ∈ {0*,* 1}*, j* = 1*, . . . , n.* (3.22)

Модели вида (3.20)–(3.22) соответствуют *задачам о покрытии множества*.

В рассмотренной задаче допускалось, что один объект может обслуживаться несколькими пожарными станциями, неравенство (3.21) гарантировало выпол- нение этого условия. Заменим в (3.21) знаки неравенства на равенства:

*n*

∑

*aijxj* = 1*, i* = 1*, . . . , m.* (3.23)

*j*=1

Получим более жесткое требование, которое означает, что нужно построить разбиение множества объектов на непересекающиеся подмножества. Модели вида (3.20), (3.22), (3.23) соответствуют задачам о разбиении множества на непересекающиеся подмножества. Заметим, что решение задачи о разбиении является также решением задачи о покрытии, но обратное неверно. Задача о разбиении может не иметь решения при разрешимой задачи о покрытии. Например, пусть *n* = 3, *m* = 3, *M*1 = 1*,* 2 , *M*2 = 1*,* 3 , *M*3 = 2*,* 3 . Тогда

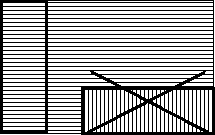
{ } { } { }

задача о покрытии имеет несколько решений: *x*1 = (1*,* 0*,* 1), *x*2 = (0*,* 1*,* 1), а

задача о разбиении не разрешима. Если же *M*3 = 3 , то задача о разбиении имеет решение: *x* = (1*,* 0*,* 1).

{ }

47



*Рис. 8.* Расположение предметов без поворотов



*Рис. 9.* Двухстадийный раскрой

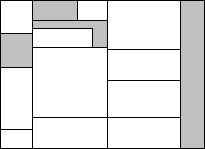
## Задача о двухстадийном гильотинном раскрое

Задачи о раскрое имеют широкое применение в различных отраслях про- мышленности: машиностроении, деревообработке, стеклопроизводстве [29]. Их суть в том, чтобы разместить и вырезать заготовки из материала с учетом различных технологических требований. Задачи классифицируются по фор- ме заготовок, их размерности, способу разрезов. Часто по технологии можно выполнять только *гильотинные разрезы*, когда каждый разрез листа выполня- ется перпендикулярно одной из сторон листа и параллельно другой стороне от одного края до другого. В зависимости от узора на поверхности определяется ориентация вырезаемых кусков. Например, если на кусках задан рисунок, то повороты могут быть запрещены (рис. 8).

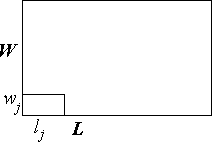
В *двухстадийной задаче* разрезы выполняются постадийно. На первой ста- дии лист разрезается по вертикали на несколько параллельных полос, на вто- рой стадии каждая полоса разрезается по горизонтали на прямоугольные куски (рис. 9).

В результате многие прямоугольники могут оказаться большего размера, чем нужно, тогда выполняется завершающая стадия по обрезанию остатков материала. На рис. 10 приводится пример двухстадийного раскроя материала с последующим обрезанием остатков.

48



*Рис. 10.* Двухстадийный раскрой с подрезкой остатков материала



*Рис. 11.* Линейные размеры для задачи о раскрое

С точки зрения оптимизации можно сформулировать различные задачи. Например, минимизировать число листов, из которых необходимо вырезать заданное количество кусков.

*Задача о двухстадийном гильотинном раскрое*

Имеется один лист материала шириной *W* , длиной *L* и *n* типов прямоуголь- ных заготовок шириной *wj* и длиной *lj* каждая, *j* = 1*, . . . , n* (рис. 11). Каждую заготовку можно вырезать в любом количестве. Известен доход *pj* от прода- жи заготовки *j*. Требуется определить, какие заготовки и в каком количестве нужно вырезать, чтобы получить максимальный доход. Повороты прямоуголь- ников при выкраивании запрещены, раскрой листа — гильотинный.

Поскольку раскрой гильотинный, значит на первой стадии лист разрезает- ся по вертикали. Пусть *k* — максимальное число вертикальных полос, тогда *k* можно оценить сверху величиной ⌊ *L* ⌋, где *lmin* — длина самого короткого прямоугольника, *lmin* = *minj*∈{1*,...,n*} *lj* . Пусть *mj* — максимальное количе- ство прямоугольников *j*-го типа, которое можно выкроить из данного листа, *mj* можно оценить сверху величиной ⌊ *LW* ⌋.

{ }

*lmin*

*lj wj*

Введем следующие целочисленные переменные:

49

*yi* — длина *i*-й полосы, *i* = 1*, . . . , k*,

*xij* — количество прямоугольников *j*-го типа, выкраиваемых из *i*-й полосы.

Чтобы выписать условие на вместимость прямоугольника *j*-го типа в *i*-ю полосу, нужно ввести вспомогательные булевы переменные:

*x*′*ij* =

1*,* если прямоугольник *j*-го типа выкраивается из *i*-й полосы,

 0 − в противном случае.



Используя введенные переменные, выпишем математическую модель. Целевая функция задачи — максимальный суммарный доход:

*k n*

∑ ∑

ограничения задачи:

max *pjxij,*

*i*=1 *j*=1

* + общая ширина всех заготовок, выкраиваемых из *i*-й полосы, не превышает ширины листа

∑

*n*

*wjxij* ≤ *W, i* = 1*, . . . , k*;

*j*=1

– суммарная длина всех полос не превышает длины листа:

*k*

∑

*yi* ≤ *L*;

*i*=1

– длина *i*-й полосы позволяет выкроить *j*-ю заготовку:

*ljx*′*ij* ≤ *yi, i* = 1*, . . . , k, j* = 1*, . . . , n.*

Необходима еще группа ограничений, чтобы связать булевы переменные *x*′*ij* с целочисленными переменными *xij* и гарантировать, что если *j*-я заготовка выкраивается из *i*-й полосы, то можно выкроить ограниченное число загото-

вок, и наоборот, нельзя выкраивать *j*-ю заготовку из *i*-й полосы, если она не помещается по размерам, т. е. нужно смоделировать два взаимоисключающих условия:

если *x*′*ij* = 0*,* то *xij* = 0*,*

если *x*′*ij* = 1*,* то *xij* ≤ *mj.*

Это можно реализовать с помощью следующих неравенств:

*xij* ≤ *mjx*′*ij , i* = 1*, . . . , k, j* = 1*, . . . , n.*

Последняя группа ограничений на значения переменных:

*xij* ≤ *mj, i* = 1*, . . . , k, j* = 1*, . . . , n,*

50

*xij* ≥ 0*,* целые, *i* = 1*, . . . , k, j* = 1*, . . . , n,*

*x*′*ij* ∈ {0*,* 1}*, i* = 1*, . . . , k, j* = 1*, . . . , n,*

*yi* ∈ {1*,* 2*, . . . , L*}*, i* = 1*, . . . , k.*

## Задача о разрезе балок

В деревообрабатывающий цех поступил заказ вырезать короткие бруски прямоугольной формы из длинных прямоугольных балок. Высота балок и брус- ков одинаковая. Заказ состоит из *b* разных типов брусков, в количестве *ni* штук каждого, *i* = 1*, . . . , b*. На складе имеется *l* разных типов балок длиной *Lj* каж- дая в количестве *Qj*, *j* = 1*, . . . , l*. Бруски можно вырезать из имеющихся балок, в этом случае стоимость выполнения одного разреза составляет *c* y. e., можно купить готовые по цене *pi* y. e. за один брусок *i*-го типа. Задача заключается в том, чтобы выполнить заказ на бруски с минимальными суммарными затра- тами.

Особенность этой задачи в том, что, прежде чем выписать математическую модель, нужно составить всевозможные схемы (варианты) раскроя балок или *карт раскроя*, учитывая длину балок *Lj*. Причем из всевозможных карт рас- кроя исключим нерациональные варианты, т. е. те схемы, которые порождают большие отходы. Под большими отходами подразумеваются такие излишки ма- териала, из которых можно было бы выкроить бруски.

Пусть *sj* — количество возможных вариантов раскроя *j*-й балки, *kj* — коли-

*is*

чество брусков *i*-го типа, выкраиваемых из *j*-го типа балки по *s*-му варианту, *f j*

*s*

— столько разрезов выполняется на *j*-й балке по *s*-му варианту, где *j* = 1*, . . . , l i* = 1*, . . . , b s* = 1*, . . . , sj*. Обрезание остатков балки до нужной длины бруска тоже считается разрезом.

Введем неотрицательные целочисленные переменные *ys* — столько раз ис-

пользуется *s*-я карта раскроя, *s* = 1*, . . . ,* (∑*l sj*), неотрицательные целочис-

*j*=1

ленные переменные *xi* — количество купленных готовых брусков, *i* = 1*, . . . , b*. Целевая функция — минимальные суммарные затраты, которые складыва-

ются из затрат на разрезание и закупку:

*l sj b*

min ∑ ∑ *cf jys* + ∑ *pixi.*

*s*

*j*=1 *s*=1 *i*=1

Общее число балок разной длины не превосходит имеющегося количества на складе:

*sj*

∑

*ys* ≤ *Qj, j* = 1*, . . . , l.*

*s*=1

Необходимо выполнить заказ на бруски разной длины:

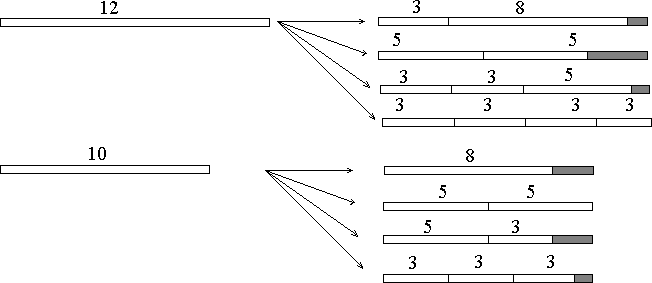
*l sj*

∑ ∑ *kj ys* + *xi* ≥ *ni, i* = 1*, . . . , b.*

*is*

*j*=1 *s*=1

51



*Рис. 12.* Возможные схемы раскроя балок

Допустимые значения переменных:

*ys* ≥ 0*,* целые*, s* = 1*, . . . , sj, j* = 1*, . . . , l, xi* ≥ 0*,* целые*, i* = 1*, . . . , b.*

Заметим, что эта модель обладает тем недостатком, что количество схем раскроя определяет количество переменных в задаче, т. е. число строк или столбцов в матрице ограничений. Количество схем раскроя конечное, но мо- жет оказаться колоссально большим, в связи с чем возникают вычислитель- ные трудности у универсальных точных методов. Однако их можно преодо- леть, пользуясь *техникой генерации столбцов* [12], которая решает задачу, не просматривая все столбцы или строки.

*Пример*

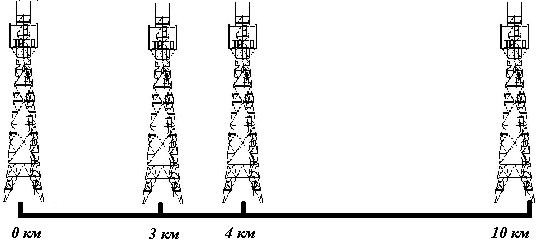
Заказ состоит из 60 брусков длиной 8 м, 40 брусков длиной 5 м и 75 брусков длиной 3 м. Длина балок составляет 12 и 10 м, в наличии имеется 20 и 25 штук соответственно. Стоимость выполнения одного разреза составляет 0,5 y.e., го- товые бруски покупаются по цене 2 y.e., 1,5 y.e., 1,1 y.e. за восьми-, пяти- и трехметровый брусок соответственно. Выполнить заказ на бруски с минималь- ными суммарными затратами.

В данном случае получается 8 карт раскроя (рис. 12). Математическая мо- дель выглядит следующим образом:

min *y*1 + *y*2 + 1*.*5*y*3 + 1*.*5*y*4 + 0*.*5*y*5 + 0*.*5*y*6 + *y*7 + 1*.*5*y*8 + 2*v*1 + 1*.*5*v*2 + 1*.*1*v*3

*y*1 + *y*2 + *y*3 + *y*4 ≤ 20*,*

52



*Рис. 13.* Расположение башен

*y*5 + *y*6 + *y*7 + *y*8 ≤ 25*, y*1 + *y*5 + *v*1 ≥ 60*,*

2*y*2 + *y*3 + 2*y*6 + *y*7 + *v*2 ≥ 40*,*

*y*1 + 2*y*3 + 4*y*4 + *y*7 + 3*y*8 + *v*3 ≥ 75*,*

*yi* ≥ 0*,* целые*, i* = 1*, . . . ,* 8*,*

*vj* ≥ 0*,* целые*, i* = 1*, . . . ,* 3*.*

## Задача о башнях

Имеется конечное множество клиентов 1*, . . . , n* и конечное множество воз- можных мест, где можно размещать башни сотовой связи 1*, . . . , m* . Извест- ны расстояния между клиентами и башнями. Каждый клиент прикрепляется к ближайшей башне, если ближайших башен несколько, то к любой из них. Нужно разместить ровно *p* башен так, чтобы после прикрепления клиентов к башням разница между максимальным и минимальным числом клиентов, закрепленных за башнями, была минимальной.

{ }

{ }

Например, пусть башни располагаются в четырех населенных пунктах вдоль дороги: в начале дороги, через 3 км, через 4 км и через 10 км от начала до- роги. Клиенты расположены в этих же пунктах (рис. 13). Нужно разместить ровно 2 башни. В табл. 5 приводится шесть возможных вариантов размещения и соответствующее значение целевой функции.

53

*Таблица 5*

Возможные варианты размещения башен

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номера  активных башен | Номер башни,  обслуживающей  соответствующий пункт  1 2 3 4 | | | | Разница  в количестве клиентов |
| (1,2) | 1 | 2 | 2 | 2 | 2 |
| (1,3) | 1 | 3 | 3 | 3 | 2 |
| (1,4) | 1 | 1 | 1 | 4 | 2 |
| (2,3) | 2 | 2 | 3 | 3 | 0 |
| (2,4) | 2 | 2 | 2 | 4 | 2 |
| (3,4) | 3 | 3 | 3 | 4 | 2 |

Сформулируем эту задачу в виде задачи целочисленного линейного програм- мирования. Введем булевы переменные:

{*y* =*j*

1*,* если в *j*-м пункте размещается башня*,*

0 − в противном случае,

1*,* если *i*-й клиент обслуживается башней в *j*-м пункте*,*

{*x* =*ij*

0 − в противном случае,

и неотрицательные целочисленные переменные:

*u* — максимальное число клиентов, обслуживаемых одной башней,

*l* — минимальное число клиентов, обслуживаемых одной башней.

Тогда в целевой функции минимизируется разница:

min(*u* − *l*)*,* (3.24)

при условиях:

*xij* ≤ *yj, i* ∈ {1*, . . . , n*}*, j* ∈ {1*, . . . , m*}*,* (3.25)

*m*

∑

*yj* = *p,* (3.26)

*j*=1

*m*

∑

*xij* = 1*, i* ∈ {1*, . . . , n*}*,* (3.27)

*j*=1

*n*

∑

*u* ≥ *xij, j* ∈ {1*, . . . , m*}*,* (3.28)

*i*=1

*n*

∑

*l* ≤ *xij* + *n*(1 − *yj*)*, j* ∈ {1*, . . . , m*}*,* (3.29)

*i*=1

54

*m*

∑

*cilxil* + (*Mi* − *cij*)*yj* ≤ *Mi, i* ∈ {1*, . . . , n*}*, j* ∈ {1*, . . . , m*}*,* (3.30)

*l*=1

и ограничениях на допустимые значения переменных:

*xij* ∈ {0*,* 1}*, i* ∈ {1*, . . . , n*}*, j* ∈ {1*, . . . , m*}*,* (3.31) *yj* ∈ {0*,* 1}*, j* ∈ {1*, . . . , m*}*,* (3.32) *u, l* ∈ Z+*,* (3.33)

где *Mi* = max*j*∈{1*,...,m*} *cij*, *i* 1*, . . . , n* .

∈ { }

Группа ограничений (3.25) уже встречалась в моделях размещения, она

гарантирует, что обслуживание клиентов будет только из «активных» ба- шен. Ограничение (3.26) насчитывает число размещенных башен. Ограничения (3.27) гарантируют, что каждый клиент обслуживается одной башней. Ограни- чения (3.28) и (3.29) определяют максимальное и минимальное число клиентов по всем башням. Для того, чтобы переменная *l* не обращалась в нуль, в правой части ограничений (3.29) добавлено слагаемое *n*(1 *yj*), которое позволяет при *yj* = 0 не ограничивать значение переменной *l* сверху, а при *yj* = 1 насчитать минимальное число клиентов и выбрать башню с наименьшей загруженностью. Ограничения (3.30) гарантируют, что каждому клиенту соответствует ближай- шая башня.

−

55

# Анализ качества моделей целочисленного линейного программирования

## Классификация моделей

В предыдущих разделах были описаны практические правила и идеи о том, как записать математическую модель в терминах *смешанного целочисленного линейного программирования*, рассмотрены разные примеры практического ха- рактера. В этом разделе продолжается рассмотрение моделей смешанного цело- численного линейного программирования, но уже с другой точки зрения. Здесь приводятся формальные определения, разные эквивалентные формы записи за- дач смешанного целочисленного линейного программирования в общем виде и теоретические аспекты оценки качества моделей. Обсуждается, как практиче- ски выбрать и построить наилучшую формулировку задачи, улучшить модель, чтобы сократить разрыв целочисленности.

Математическая модель следующего вида называется *моделью смешанного целочисленного линейного программирования, записанная в стандартной фор- ме* (сокращенно *M IP* , от англ. mixed integer programming):

min(*cx* + *f y*) (4.1)

*Ax* + *By* ≥ *b,* (4.2)

*x* ≥ 0*,* (4.3)

*y* ≥ 0*,* целые*,* (4.4)

где *c* = (*c*1*, . . . , cn*), *f* = (*f*1*, . . . , fp*) — вектор-строки с вещественными компо- нентами, задающие коэффициенты целевой функции;

*x* = (*x*1*, . . . , xn*)*T* — вектор-столбец переменных задачи, принимающих неотри- цательные вещественные значения;

*y* = (*y*1*, . . . , yp*)*T* — вектор-столбец переменных задачи, принимающих неотри- цательные целочисленные значения;

*A*, *B* – матрицы размерности (*m n*) и (*m p*) соответственно, с рациональными значениями компонент, определяющие коэффициенты в системе из *m* ограни- чений;

× ×

*b* = (*b*1*, . . . , bm*)*T* — вектор-столбец с вещественными компонентами правых ча- стей ограничений.

Обозначим через *XMIP* множество значений переменных, удовлетворяющих ограничениям (4.2)–(4.4). Множество *XMIP* называют *множеством допусти- мых* решений задачи. Обозначим через *z*∗(*XMIP* ) оптимальное значение целе- вой функции (4.1) на множестве *XMIP* .

Заметим, что в стандартной форме требуется, чтобы задача была на мини- мум, все знаки в ограничениях были нестрогими, вида , а переменные прини- мали неотрицательные значения.

≥

Существуют каноническая и общая формы записи *M IP* . Каждую из этих форм можно преобразовать в стандартную, пользуясь следующими простыми соображениями:

56

* ограничения равенства *Ax* + *By* = *b* эквивалентны паре неравенств *Ax* +

*By b* и *Ax By b*;

≥ − − ≥ −

* задача на максимум с целевой функцией (*cx* +*f y*) эквивалентна задаче на минимум с целевой функцией (*cx* + *f y*)*.* Аналогичные преобразования, т. е. умножение на ( 1), нужно сделать в неравенствах со знаком ;

− ≤

−

* каждое вхождение *свободной переменной x*, т. е. переменной без ограни- чения на знак, нужно заменить разностью двух новых неотрицательных пере- менных, *x* = (*u v*). Таким образом, все переменные в модели будут неотрица- тельными.

−

Любое оптимальное решение, полученное для модели в стандартной форме, нетрудно преобразовать в оптимальное решение исходной задачи, и наоборот.

Задача *M IP* , в которой все переменные принимают значения только 0 или 1, называют *задачей булева, или 0–1, программирования*.

Задача *M IP* , в которой все целочисленные переменные принимают значе- ния 0 или 1, называют *задачей смешанного 0–1 программирования*.

Задача *M IP* , в которой все переменные принимают вещественные значе- ния, называют *задачей линейного программирования* (сокращенно *LP* , от англ. linear programming).

Задача *M IP* , в которой все переменные принимают только целые значения, называют *задачей полностью целочисленного программирования* (сокращенно *IP* , от англ. integer programming).

Известно, что задача *IP* является NP-трудной в общем случае [5], но поли- номиально разрешима при фиксированном числе переменных [22]. Задача *IP* имеет широкие рамки, внутри которых формулируются задачи *M IP* смешан- ного 0–1 программирования. Слабое место этих задач в том, что в общем виде не известно алгоритма, практически применимого для решения задач большой размерности.

Решать задачу *LP* в общем случае проще, чем соответствующую задачу *M IP* , потому что для *LP* существуют точные полиномиальные алгоритмы [13]. Поэтому иногда в задачах *M IP* отказываются от условий целочисленности и решают соответствующую задачу *LP* . Некоторые задачи обладают тем свой- ством, что оптимальное решение задачи *LP* является целочисленным, однако это не всегда так. Отказ от условий целочисленности в задаче *M IP* и переход к соответствующей задаче *LP* называют *линейной релаксацией*. Под линейной релаксацией задачи (4.1)–(4.4) понимается следующая задача:

|  |  |
| --- | --- |
| min(*cx* + *f y*) | (4.5) |
| *Ax* + *By* ≥ *b,* | (4.6) |
| *x* ≥ 0*,* | (4.7) |
| *y* ≥ 0*.* | (4.8) |
| 57 |  |

Будем обозначать задачу (4.5)–(4.8) через *LR* (от англ. relaxation — ослабле- ние), а множество допустимых решений (4.6)–(4.8) через *XLR*. Нетрудно заме- тить, что множество *XLR* шире допустимого множества *XMIP* : *XMIP XLR*. Значит, оптимальное значение целевой функции задачи линейной релаксации дает нижнюю оценку на оптимум соответствующей задачи *M IP* : *z*∗(*XLR*)

⊂

≤

*z*∗(*XMIP* ), если исходная задача на минимум, и дает верхнюю оценку, если

исходная задача *M IP* на максимум.

## Разрыв целочисленности

Величина отклонения оптимального значения линейной релаксации от оп- тимального значения целевой функции для задачи смешанного целочисленного программирования называется *разрывом целочисленности*. Рассматривают *аб- солютный разрыв целочисленности*

|*z*∗(*XLR*) − *z*∗(*XMIP* )|

и *относительный разрыв целочисленности*

|*z*∗(*XLR*) − *z*∗(*XMIP* )|

max{|*z*∗(*XLR*)|*,* |*z*∗(*XMIP* )|}

* 100%*.*

Для вычисления разрыва нужно точно решить обе задачи. Часто вместо оп- тимальных значений рассматривают хорошие приближенные значения для *z*∗(*XLR*) и *z*∗(*XMIP* ) и тогда вместо разрыва целочисленности вычисляют сле- дующую величину:

*UB* − *LB* 100%*,*

·

*U B*

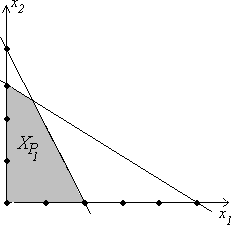
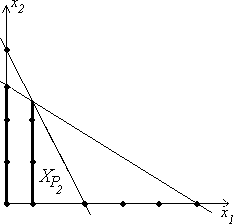
где *U B*, *LB* — верхняя и нижняя оценка оптимального решения соответствен- но. Оптимальное значение должно лежать в промежутке [*LB, U B*]. Иногда в решателях разрыв вычисляется по отношению к нижней оценке, т. е. в знаме- нателе стоит *LB* вместо *U B*.

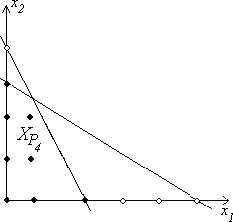
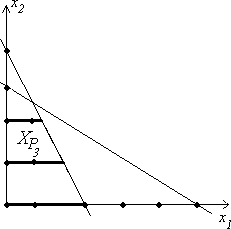
Рассмотрим несколько примеров, чтобы наглядно увидеть, к каким измене- ниям в оптимальном решении приводит добавление требований целочисленно- сти переменных.

*Пример* [28]

Рассмотрим следующую задачу *P*1:

|  |  |
| --- | --- |
| max(*x*1 + *x*2) | (4.9) |
| 3*x*1 + 5*x*2 ≤ 15*,* | (4.10) |
| 5*x*1 + 2*x*2 ≤ 10*,* | (4.11) |
| *x*1 ≥ 0*,* | (4.12) |
| *x*2 ≥ 0*.* | (4.13) |
| 58 |  |



*Рис. 14.* Допустимые области для задач *P*1, *P*2, *P*3, *P*4

Заменим в задаче *P*1 условие (4.12) на условие *x*1 0*, целые* и обозначим новую задачу *P*2. Заменим в задаче *P*1 условие (4.13) на условие *x*2 0*, целые* и обозначим новую задачу *P*3. Заменим в задаче *P*1 оба условия (4.12), (4.13) на *x*1 0*, целые* и *x*2 0*, целые*, полученную задачу *IP* обозначим *P*4.

≥

≥

≥ ≥

На рис. 14 изображено графическое представление допустимой области для каждой из задач *P*1, *P*2, *P*3 и *P*4.

Задача *P*1 является задачей линейного программирования, и ее допустимая

область *XP*1 — это заштрихованная часть, в том числе и границы. Для задачи смешанного целочисленного программирования *P*2 допустимая область *XP*2 — это вертикальные отрезки, проходящие через точки с целочисленными значе- ниями *x*1, удовлетворяющие ограничениям задачи. Аналогично для задачи *P*3 допустимая область *XP*3 — это горизонтальные отрезки. Допустимая область *XP*4 для задачи полностью целочисленного программирования *P*4 состоит из восьми точек с целочисленными координатами, удовлетворяющих ограниче-

59

ниям задачи. Таким образом, после добавления требования целочисленности область *XP*4 стала существенно меньше по сравнению с *XP*1 , *XP*2 , *XP*3 *.* Рас- смотрим оптимальные решения этих задач:

*P*1 : *z*∗(*XP*1 ) = 3*.*4211 *x*∗ = (1*.*0526*,* 2*.*3684);

*P*2 : *z*∗(*XP*2 ) = 3*.*4 *x*∗ = (1*,* 2*.*4);

*P*3 : *z*∗(*XP*3 ) = 3*.*2 *x*∗ = (1*.*2*,* 2);

*P*4 : *z*∗(*XP*4 ) = 3 *x*∗ = (1*,* 2)*.*

Нетрудно заметить, что, переходя от решения задачи *P*1 с наименее жест- кими требованиями к задаче *P*4 с более жесткими требованиями, оптимальное значение целевой функции уменьшается: *z*∗(*XP*4 ) *< z*∗(*XP*1 ).

*Пример*

В рассмотренных выше примерах оптимальное решение задачи *P*4 может быть получено округлением решения задачи *P*1 вниз. Однако округление не всегда приводит к оптимальному решению соответствующей задачи *M IP* . Рас- смотрим следующую задачу:

max(2*x*1 + *x*2) (4.14)

7*x*1 + 48*x*2 ≤ 84*,* (4.15)

−*x*1 + 12*x*2 ≥ 3*,* (4.16)

*x*1 ≥ 0*,* целые, (4.17)

*x*2 ≥ 0*,* целые. (4.18)

Оптимальным решением соответствующей задачи линейного программи- рования будет *x*∗*LP* = (6*.*5455*,* 0*.*7955) со значением целевой функции *zL*∗ *P* = 13*.*8864. Оптимальным решением исходной задачи является вектор *x*∗*IP* = (5*,* 1) со значением целевой функции *zI*∗*P* = 11. Из рис. 15 видно, что *x*∗*IP* нельзя получить простым округлением решения *x*∗*LP* .

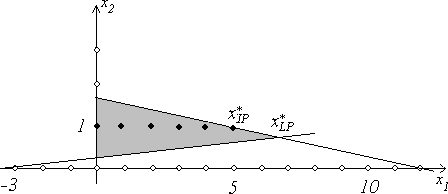
Вычислим разрыв целочисленности. Абсолютный разрыв составляет

13*.*8864 11 = 2*.*8864, относительный равен 2*.*8864 100 % = 20*.*79 %. Величина относительного разрыва дает информацию о том, насколько сложной в вычис- лительном плане окажется задача для точных методов, которые используют решение линейной релаксации при доказательстве оптимальности целочислен- ного решения. Обычно задачи с относительным разрывом 10 % считаются про- стыми, а задачи с разрывом 50 % — сложными. В связи с этим среди разных возможных математических моделей *M IP* лучше выбирать ту, у которой раз- рыв целочисленности меньше.

13*.*8864

− ·

60



*Рис. 15.* Оптимальные целочисленное и дробное решения

Более того, может оказаться, что задача *LP* имеет решение, а соответству- ющая задача *IP* не разрешима. Например, заменим ограничение (4.16) на

−*x*1 + 12*x*2 ≥ 13*,* (4.19)

тогда область допустимых решений в задаче (4.14), (4.15), (4.17)–(4.19) — это треугольник *ABC* (рис. 16). Оптимальное решение соответствующей задачи *LP* находится в точке *C*. Заметим, что ни одна целочисленная точка не входит в область *ABC*, следовательно, множество допустимых решений в задаче *IP* (4.14), (4.15), (4.17)–(4.19) является пустым.

*Пример*

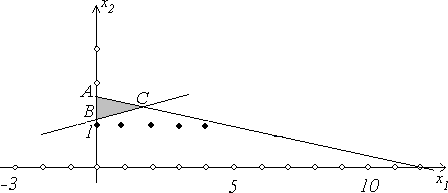
Пусть допустимая область образована неравенствами:

|  |  |
| --- | --- |
| *x*1 + *x*2 ≤ 10*,* | (4.20) |
| 5*x*1 + 3*x*2 ≥ 15*,* | (4.21) |
| *x*1 ≤ 6*,* | (4.22) |
| *x*2 ≤ 7*,* | (4.23) |

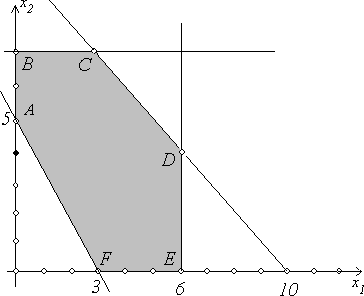
*x*1 ≥ 0*, x*2 ≥ 0*.* (4.24)

Если изобразить графически, то допустимая область — это многогранник *ABCDEF* (рис. 17). Из теории линейного программирования известно, что оптимальное решение задачи *LP* лежит в одной из угловых точек допустимой области. В данном примере все угловые точки имеют целочисленные координа- ты. Следовательно, при любой целевой функции среди оптимальных решений задачи линейного программирования на этом множестве всегда будет целочис- ленное решение. В этом случае разрыв целочисленности равен нулю.

61



*Рис. 16.* Допустимая область — треугольник *ABC*



*Рис. 17.* Допустимая область — многогранник *ABCDEF*

62

*Пример. Задача планирования производства*

Завод занимается производством некоторой однотипной продукции. Про- дукция производится в начале месяца и либо отправляется потребителям сразу, либо хранится на складе. Продукцию можно производить и хранить на скла- де на протяжении всего горизонта планирования. Производственные затраты складываются из единовременных расходов на запуск производства, удельных производственных затрат и удельных затрат на хранение продукции. Задан промежуток времени *T* месяцев, в течение которого необходимо спланировать производство и хранение продукции так, чтобы выполнить заказ с минималь- ными затратами.

Обозначим через *dt* — заказ на продукцию в месяц *t*, *ct* — затраты на запуск производства в месяц *t*, *pt* — удельные затраты на производство в месяц *t*, *ht*

— удельные затраты на хранение продукции в течение месяца *t*. Выпишем две модели с разными переменными.

*Первая модель* использует следующие переменные:

*yt* — количество продукции, произведенной в месяц *t*;

*st* — количество продукции, оставленной на складе к концу месяца *t*;

{*x* =*t*

1*,* если в месяц *t* было запущено производство*,*

0 в противном случае.

−

Тогда получаем следующую математическую модель:

*T*

∑

min (*ctxt* + *ptyt* + *htst*) (4.25)

*t*=1

*y*1 = *d*1 + *s*1*,* (4.26)

*st*−1 + *yt* = *dt* + *st, t* = 2*, . . . , T,* (4.27)

*T*

∑

*yt* ≤ *dkxt, t* = 1*, . . . , T,* (4.28)

*k*=1

*sT* = 0*,* (4.29)

*yt* ≥ 0*, st* ≥ 0*,* целые*, xt* ∈ {0*,* 1}*, t* = 1*, . . . , T.* (4.30)

Целевая функция (4.25) — минимальные общие затраты, которые складывают- ся из затрат на запуск, производство и хранение. Ограничения (4.26) и (4.27) отвечают за удовлетворение спроса и выполнение «закона сохранения» про- дукции в каждом месяце. Поскольку в начальный момент времени на складе отсутствует продукция, то для первого месяца выписано отдельное условие. Ограничения (4.28) устанавливают связь между переменными *xt* и *yt* и огра- ничивают сверху максимальный ежемесячный объем партии. Только если про- изводство запущено в месяц *t*, на нем может производится продукция, но не более, чем величина суммарного спроса на него. Ограничение (4.29) задает уровень запасов на складе в конце горизонта планирования.

63

*Вторая модель* использует следующие переменные: *qit* — столько продукции производится в месяц *i*, чтобы выполнить заказ в месяц *t*, *t i*, переменные *xt* остались теми же, что и в предыдущей модели. Используя введенные перемен- ные, получаем следующую математическую модель:

≥

*T t T*

min ∑ ∑(*pi* + *hi* + *hi*+1 + *. . .* + *ht*−1)*qit* + ∑ *ctxt,* (4.31)

*t*=1 *i*=1

*t*=1

*t*

*qit* = *dt, t* = 1*, . . . , T,* (4.32)

∑

*i*=1

*qit* ≤ *dtxi, i* = 1*, . . . , T, t* = *i, . . . , T,* (4.33)

*qit* ≥ 0*, xt* ∈ {0*,* 1}*, t* = 1*, . . . , T.* (4.34)

Вторая модель обладает тем свойством, что если требование целочисленности переменных *xt* заменить на условие непрерывности переменных 0 *xt* 1, то в оптимальном решении соответствующей задачи линейной релаксации пере- менные *xt* будут принимать значение 0 или 1, т. е. разрыв целочисленности для *LR* (4.31)–(4.34) равен нулю. Первая модель — это *M IP* , поскольку в ограниче- ниях (4.28) имеются большие константы, то разрыв целочисленности не равен нулю.

≤ ≤

Следующая теорема дает оценку близости решений *IP* и соответствующей

*LR*.

***Теорема*** [22]***.*** *Пусть A — целочисленная* (*m n*)*-матрица, каждый минор которой по абсолютной величине не превосходит* ∆*, и пусть даны векторы b и c. Предположим, что оба максимума*

×

max{*cx*|*Ax* ≤ *b*} (4.35)

*и*

*конечны. Тогда:*

max{*cx*|*Ax* ≤ *b, x* ∈ } (4.36)

1. *Для любого оптимального решения y задачи (4.35) существует опти- мальное решение z задачи (4.36), для которого y z* ∞ *n*∆*.*

∥ − ∥ ≤

1. *Для любого оптимального решения z задачи (4.36) существует опти- мальное решение y задачи (4.35), для которого* ∥*y* − *z*∥∞ ≤ *n*∆*.*

( *y* ∞ обозначает максимум абсолютных величин компонент вектора *y*.)

∥ ∥

Эта теорема обобщает результат о том, что для любой рациональной мат-

рицы *A* существует такой конечный набор векторов *x*1*, . . . , xN* , что для любой вектор-строки *c* и любого целочисленного вектор-столбца *b* из того, что *x*∗ яв- ляется оптимальным базисным решением (вершиной) релаксированной задачи

целочисленного линейного программирования max{*cx*|*x* ≥ 0*, Ax* = *b*}, следует,

64

что оптимум *IP* max *cx x* 0*, Ax* = *b, x* Z достигается на одном из векторов

{ | ≥ ∈ }

(*x*∗ *xk*), *k* = 1*, . . . , N* .

−

*Пример*

Следующий пример показывает, что указанная в теореме граница *n*∆ не улучшаема.

Пусть



+1 0 0 · · · 0

−1 +1 0 · · · 0

*A* :=  0 −1 +1 · · · 0

 *,*





. . . . . 

0 *. . .* 0 −1 +1





*β β*

 .

*b* :=

.

*β*

 *,*

*c* := (1*, . . . ,* 1)

для любого *β*, 0 ≤ *β <* 1 следующие векторы являются единственными реше- ниями задач max{*cx*|*Ax* ≤ *b*} и max{*cx*|*Ax* ≤ *b, x* ∈ Z}:



*β*

2*β*

*y* :=

 .

.

*nβ*

 *,*

 0 

0

 .



и ∥*y* − *z*∥∞ ≤ *n*∆*β*.

*z* :=

.

0

## Число ограничений и переменных в модели

Качество построенной математической модели можно оценивать по разным показателям. Для задач булевого программирования одним из показателей яв- ляется количество булевых переменных. Число переменных особенно важно при решении задач точными методами, одним из которых является метод вет- вей и границ [13]. В процессе решения этим методом строится так называемое дерево ветвления. Число вершин в дереве зависит от количества булевых пе- ременных. Таким образом, если в модели *n* булевых переменных, то в дереве

65

ветвления может быть 2*n* вершин в худшем случае. Следовательно, с ростом *n*

время решения задачи может расти экспоненциально.

Рассмотрим на примерах несколько идей для сокращения количества пере- менных.

*Пример*

Имеется парк из трех грузовиков и два маршрута, по которым нужно от- править грузовики. Предполагается, что все грузовики одинаковые. Требуется назначить грузовики на маршруты. Это можно сделать несколькими способа- ми.

*Первый способ.* Пусть

{*x* =*ij*

1*,* если грузовик *i* отправляется по маршруту *j,*

0 − в противном случае,

где *i* 1*,* 2*,* 3 , *j* 1*,* 2 . Этот способ может привести к тому, что в дереве ветвления в худшем случае будет 26 вершин.

∈ { } ∈ { }

*Второй способ.* Пусть

{*y* =*i*

1*,* если грузовик *i* отправляется по первому маршруту,

0*,* если грузовик *i* отправляется по второму маршруту,

где *i* 1*,* 2*,* 3 . Этот способ может привести к тому, что в дереве ветвления в худшем случае будет 23 вершин.

∈ { }

Второй способ приводит к меньшему количеству переменных для ветвления.

Однако оба способа обладают недостатком, обусловленным наличием симмет- рии в решениях, а именно: поскольку все грузовики одинаковые, то пара допу- стимых решений (первый способ)

*x*11 = 1*, x*22 = 1*, x*32 = 1 (4.37)

и

*x*12 = 1*, x*21 = 1*, x*32 = 1*,* (4.38)

и пара допустимых решений (второй способ)

*y*1 = 1*, y*2 = 0*, y*3 = 0 (4.39)

и

*y*1 = 0*, y*2 = 1*, y*3 = 0 (4.40)

по сути, представляют одно и то же назначение грузовиков на маршруты. Один из грузовиков обслуживает один маршрут, а два других грузовика обслужива- ют другой маршрут. В данном примере можно избежать симметрии и одновре- менно сократить число переменных введением других целочисленных перемен- ных.

66

*Третий способ.* Пусть *nj* — количество грузовиков, отправленных по марш- руту *j*. Тогда решение

*n*1 = 1*, n*2 = 2

и решения (4.37), (4.38), (4.39) и (4.40), по сути, подразумевают одно и то же назначение грузовиков на маршруты, но теперь в модели, в три раза меньше переменных.

Однако в данном случае в модели нет булевых переменных, и процесс ветв- ления по целочисленным переменным в методе ветвей и границ будет прохо- дить совсем иначе. Таким образом, не всегда модель с меньшим количеством переменных лучше.

Одним из преимуществ модели может быть наличие переменной, выбирая которую можно реализовать дихотомию при ветвлении по множеству решений. Поясним это на следующем примере.

*Пример*

Планируется открытие фабрики. Фабрика может размещаться на юге или на севере страны. Кроме того, нужно определить, будет ли конечный продукт упаковываться непосредственно на фабрике или нет.

Введем булевы переменные:

 1*,* если фабрика открывается на юге

*xsp* =

*xsn* =



и упаковывает продукт,

0 − в противном случае,





1*,* если фабрика открывается на юге и не упаковывает продукт,

*xnp* = 



0 − в противном случае,

1*,* если фабрика открывается на севере и упаковывает продукт,

0 − в противном случае,

*xnn* =

1*,* если фабрика открывается на севере и не упаковывает продукт,

 0 − в противном случае.

Ограничение-равенство

*xsp* + *xsn* + *xnp* + *xnn* = 1 (4.41)

гарантирует, что будет реализован только один вариант работы фабрики.

Однако при таких переменных и ограничении процесс дихотомии при ветв- лении, соответствующий тому, размещается ли фабрика на юге или на севере, на одной переменной организовать не удастся. Поэтому разумно ввести новую вспомогательную булеву переменную:

{*y* =

1*,* если фабрика открывается на севере,

0*,* если фабрика открывается на юге.

67

Тогда вместо ограничения (4.41) будет пара ограничений:

*xnp* + *xnn* − *y* = 0

и

*xsp* + *xsn* + *y* = 1*.*

Аналогично можно было ввести другую булеву переменную, чтобы ветвление соответствовало тому, будет ли продукт упаковываться или нет.

*Пример* [31]

Пусть в задаче имеется ограничение вида

∑

*ajxj* ≤ *b,*

*j*

где все переменные *xj*, коэффициенты *aj* и правые части ограничений *b* — целые числа.

В процессе решения программными средствами это ограничение приводит- ся к канонической форме. Для этого добавляется так называемая переменная недостатка *u*, *u* ≥ 0, такая, что

∑

*ajxj* + *u* = *b.* (4.42)

*j*

Однако это можно сделать вручную на этапе построения модели. Нетрудно за- метить, что если все коэффициенты и переменные в ограничении целые числа, то *u* может принимать только целые значения. Наличие этой переменной поз- воляет организовать процесс ветвления по ней, а само ограничение (4.42) будет выступать в качестве отсекающей плоскости в соответствующей задаче *LR* [22].

## Многогранники. Правильные неравенства

Рассмотрим следующую задачу *IP* :

min *f x Ax* ≤ *b,*

*x* ≥ 0*,* целые*.*

Пусть *X* — множество допустимых решений этой задачи. Множество точек *P* = *x Rn Ax b* , удовлетворяющих конечному числу линейных неравенств, называют *многогранником*.

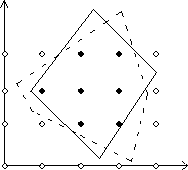
{ ∈ | ≤ }

Многогранник *P* называют *представлением* множества допустимых реше- ний *X*, если *X* совпадает с целочисленными точками из многогранника, т. е. *X* = *P* Z*n.* На рис. 18 приводится два представления.

∩

Пусть *P* 1 и *P* 2 — два представления для *X*. Как уже неоднократно наблю- далось в предыдущих разделах, для одной и той же задачи можно предложить

68



*Рис. 18.* Два представления

несколько математических моделей. Возникает вопрос: «Существует ли наи- лучшее представление и как его найти?»

*Представление P* 1 *лучше P* 2, если *P* 1 *P* 2, поскольку допустимые цело- численные решения задачи лежат и в *P* 1, и в *P* 2, но с точки зрения линей- ной релаксации допустимая область представления *P* 2 больше, поэтому раз- рыв целочисленности с *P* 1 будет меньше, чем с *P* 2, и для задачи на минимум *z*∗(*X*) *z*∗(*P* 1) *z*∗(*P* 2)*.*

⊂

≥ ≥

*Выпуклой оболочкой множества X* называется множество *conv*(*X*), состо-

ящее из точек вида *x* = ∑*T λixi*, где ∑*T λi* = 1, *λi* ≥ 0, *i* = 1*, . . . , T* , где

{ }

*i*=1

*i*=1

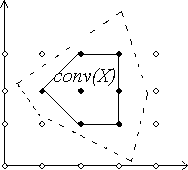
{ }

*x*1*, . . . , xT* все точки из *X*, а *T* — общее число точек в множестве *X*. В этом случае точка *x* является выпуклой комбинацией точек *x*1*, . . . , xT* . Выпуклая оболочка *conv*(*X*) образует многогранник и является лучшим из возможных представлений для *X* (рис. 19). Суть выпуклой оболочки в том, что все край- ние точки допустимого множества *X* содержатся в ней. Из теории линейного программирования известно, что оптимальное решение задачи *LP* достигается именно в крайних точках. Кроме того, *conv*(*X*) содержится в любом множе- стве, содержащем *X*. Таким образом, *conv*(*X*) — это минимальное множество, содержащее *X* и описываемое системой линейных неравенств, а значит модель задачи с представлением, в котором оно совпадает с выпуклой оболочкой, явля- ется самой лучшей моделью с точки зрения разрыва целочисленности. В этом случае разрыв равен нулю.

С практической точки зрения поиск такой переформулировки задачи стал- кивается со следующими трудностями:

1. заранее неизвестно, как задать *conv*(*X*), а ее нахождение является не ме- нее сложной задачей, чем решить исходную задачу *M IP* ;
2. для многих задач *M IP* описание *conv*(*X*) требует экспоненциального чис-

69



*Рис. 19.* Наилучшее представление

ла переменных и линейных неравенств, а значит для решения полученной таким образом задачи *LP* потребуется много времени. С другой стороны, добавление «правильных» ограничений улучшает верхнюю оценку *LP* и может заметно ускорить время работы алгоритма. Неравенства, добав- ление которых не меняет целочисленную допустимую область, называют *правильными неравенствами*.

*Пример*

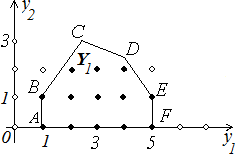
Пусть допустимая область *Y*1 задается следующими неравенствами:

|  |  |
| --- | --- |
| *y*1 ≥ 1*,* | (4.43) |
| *y*1 ≤ 5*,* | (4.44) |
| *y*1 + 0*.*8*y*2 ≤ 5*.*8*,* | (4.45) |
| *y*1 − 0*.*8*y*2 ≥ 0*.*2*,* | (4.46) |
| *y*1 + 8*y*2 ≤ 26*,* | (4.47) |
| *y*1 ≥ 0*,* целые, | (4.48) |
| *y*2 ≥ 0*,* целые. | (4.49) |

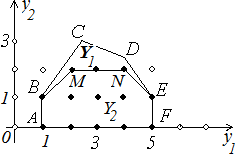
На рис. 20 область *Y*1 соответствует точкам с целочисленными координата- ми, лежащим внутри многогранника *ABCDEF* . В задаче линейной релак- сации допустимая область представляет внутреннюю область многогранника *ABCDEF* . Рассмотрим область *Y*2, которая формируется следующей системой неравенств:

*y*1 + *y*2 ≤ 6*,* (4.50)

70



*Рис. 20.* Многогранник *ABCDEF*



*Рис. 21.* Целочисленные допустимые решения

*y*1 − *y*2 ≥ 0*,* (4.51)

*y*2 ≤ 2*,* (4.52)

*y*1 ≥ 0*,* (4.53)

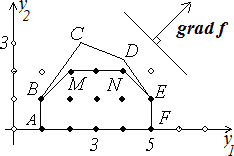
*y*2 ≥ 0*.* (4.54)

Заметим, что множество целочисленных точек *Y*1 и *Y*2 совпадает (рис. 21) Следовательно, неравенства (4.50)–(4.52) — правильные.

Добавление правильных неравенств позволяет сузить допустимую область в задаче линейной релаксации, но не меняет допустимой области в задаче цело- численного программирования. На рис. 21 видно, что после добавления нера- венств (4.50)–(4.52) к системе (4.43)–(4.49) допустимая область задачи линей- ной релаксации уменьшилась и совпадает с внутренней частью многогранника *ABM N EF* .

Пусть *f* — линейная целевая функция. На рис. 22 показано направление градиента *f* . Требуется найти max *f* при ограничениях (4.43)–(4.49). Мы уста-

71



*Рис. 22.* Допустимая область и направление градиента

новили, что добавление неравенств (4.50)–(4.52) не меняет целочисленную об- ласть, следовательно, вместо исходной задачи можно решать задачу линейной релаксации (4.43)–(4.52). Оптимальное решение задачи линейной релаксации лежит в крайней точке *N* с целочисленными координатами. Таким образом, разрыв целочисленности равен нулю. Это означает, что оптимальное решение задачи *M IP* с целевой функцией *f* при ограничениях (4.43)–(4.49) совпадает с оптимальным решением задачи *LP* с целевой функцией *f* при ограничениях (4.43)–(4.52). Следовательно, в данном случае вместо задачи *M IP* можно ре- шать задачу *LP* . Допустимая область, образованная (4.43)–(4.52), совпадает с *conv*(*Y*1 *Y*2).

∪

В рассмотренном выше примере неравенство *y*1 + *y*2 6 — правильное,

≤

но оно не улучшает представление допустимого множества с точки зрения со- кращения разрыва целочисленности. Неравенство *y*2 2 не только является правильным, но еще и формирует *conv*(*Y*1). Возникает вопрос: «Как строить правильные неравенства, определяющие выпуклую оболочку?»

≤

*Построение правильных неравенств*

Существует много способов построения правильных неравенств [25, 30], но универсального алгоритма не известно. Рассмотрим один из наиболее простых способов.

В общем случае под релаксацией множества *X* понимается любое надмноже- ство *Y* , такое, что *X Y* . Тогда любое правильное неравенство для множества *Y* является правильным неравенством для множества *X*. В качестве релакса- ции можно взять часть неравенств, определяющих область *X* и ограничения на значения переменных.

⊂

*Пример*

Вернемся к множеству *Y*1, которое задается системой неравенств (4.43)– (4.49), и построим правильное неравенство. Мы уже заранее знаем ответ, это неравенство (4.50), которое обсуждалось выше.

Рассмотрим в качестве релаксации множества *Y*1 множество *Y* , состоящее из

72

точек, удовлетворяющих следующим неравенствам: (4.43)—(4.45), 0 *y*2 3, (4.48) и (4.49). Неравенство *y*2 3 получается, если учесть, что *y*1 +8*y*2 26 и *y*1 0, *y*1 целые, а следовательно, *y*2 26 = 3. Теперь, учитывая (4.45), за- метим, что неравенство *y*1 +*y*2 6 является правильным для *Y* , следовательно, оно будет правильным и для *Y*1.

8

≤

≥ − ≤ ⌊ ⌋

≤ ≤

≤ ≤

Большинство программных средств, прежде чем решать задачу, автомати- чески генерируют подобного рода правильные неравенства и добавляют их в исходную задачу. Заметим, что получение правильных неравенств не требует специальных знаний о структуре исходного множества допустимых решений.

*Пример. Правильные неравенства для задачи планирования производства*

Вернемся к задаче (4.25)–(4.30). Рассмотрим правильные неравенства для этой задачи.

*Утверждение*. Для любых *l* и *C* неравенства

∑ *yi* ≤ ∑(∑ *dt*)*xi* + *sl,* 1 ≤ *l* ≤ *T, C* ⊆ {1*, . . . , l*} (4.55)

*i*∈*C*

*i*∈*C*

*t*=*i*

*i*∈*C*

*i*∈*C*

*t*=*i*

*l*

являются правильными для задачи (4.25)–(4.30).

*Доказательство*. Проверим, что добавление неравенств (4.55) в модель не приводит к потере допустимых решений задачи. Возьмем допустимое решение (*y, s, x*) задачи (4.26)–(4.30) и покажем, что неравенства (4.55) выполняются. Рассмотрим два случая.

1. *xi* = 0 для любого *i* ∈ *C*.

Тогда *yi* = 0 для любого *i* ∈ *C* и из (4.55) получаем, что *sl* ≥ 0.

1. *xi* = 1 для некоторого *i* ∈ *C*.

Пусть *k* = min{*i* ∈ *C*|*xi* = 1}*.* Тогда *yi* = 0 для любого *i < k* и

∑

∑(

*t*=*k*

*t*=*k*

∑*i*∈*C yi* ≤

*l t*=*k*

*yt* =∑*l*

*dt* + *sl* − *sk*−1 ≤ ∑*l*

*dt* + *sl* ≤ ∑

*i*∈*C*

*l t*=*i*

*dt*)*xi* + *sl.* Что и

требовалось доказать.

Известен и более сильный результат.

***Теорема*** [30]***.*** *Выпуклая оболочка множества* (4.26)–(4.30) *задается сле- дующей системой ограничений:*

*y*1 = *d*1 + *s*1*,* (4.56)

*st*−1 + *yt* = *dt* + *st, t* = 2*, . . . , T,* (4.57)

*sT* = 0*,* (4.58)

*xt* ≤ 1*, t* = 1*, . . . , T,* (4.59)

*l*

∑ *yi* ≤ ∑(∑ *dt*)*xi* + *sl для любого l,* и *C* ∅*,* (4.60)

*i*∈*C*

*i*∈*C*

*t*=*i*

*i*∈*C*

*i*∈*C*

*t*=*i*

*yt* ≥ 0*, st* ≥ 0*, xt* ≥ 0*, t* = 1*, . . . , T.* (4.61)

73

## Целочисленные решения задачи линейного программирова- ния

Структура некоторых задач, например, задачи о назначениях, задачи о на- хождении потока минимальной стоимости и других, такова, что оптимальное решение соответствующей задачи линейного программирования является цело- численным. Это важная характеристика задачи. Если все решения задачи ли- нейного программирования — целочисленные, то обычный аппарат линейного программирования позволяет найти оптимальное решение, удовлетворяющее требованию целочисленности и являющееся, следовательно, также оптималь- ным решением соответствующей задачи целочисленного линейного програм- мирования. Причина такой благоприятной ситуации может быть в том, что система ограничений обладает определенными свойствами, о которых сейчас пойдет речь.

Квадратная матрица *D* = (*dij*) с целочисленными компонентами, *i* = 1*, . . . , m*, *j* = 1*, . . . , m* называется *унимодулярной*, если ее определить равен

1.

±

Матрица *A* = (*aij*) с целочисленными компонентами, *i* = 1*, . . . , m*, *j* = 1*, . . . , n* называется *полностью унимодулярной*, если определитель любой ее квадратной подматрицы равен 0 или 1.

±

Таким образом, полностью унимодулярная матрица состоит из 0, 1 или 1.

−

Однако, например, матрица

*A* =  

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |

не является полностью унимодулярной, поскольку определитель подматрицы, составленной из первых трех столбцов и первых трех строк, равен —2.

К сожалению, определение не дает эффективного способа проверки, являет- ся ли матрица полностью унимодулярной, так как вычисление определителей всех квадратных подматриц матрицы *A* — трудоемкая процедура. Следующая теорема, которая является непосредственным следствием из определения, опи- сывает варианты построения других унимодулярных матриц из заданной.

***Теорема*** [30]***.*** *Следующие утверждения эквиваленты.*

1. *Матрица A является полностью унимодулярной.*
2. *Матрица, полученная транспонированием матрицы A, является полно- стью унимодулярной.*
3. *Матрица, полученная приписыванием к A единичной матрицы E справа, является полностью унимодулярной.*
4. *Матрица, полученная удалением строки (столбца) из матрицы A, яв- ляется полностью унимодулярной.*

74

1. *Матрица, полученная умножением строки (столбца) матрицы A на*

(−1)*, является полностью унимодулярной.*

1. *Матрица, полученная перестановкой двух строк (столбцов) матрицы*

*A, является полностью унимодулярной.*

1. *Матрица, полученная дублированием строк (столбцов) матрицы A, яв- ляется полностью унимодулярной.*
2. *Матрица, полученная после выполнения операции замены базисной пере- менной, является полностью унимодулярной.*

Следующее достаточное условие гарантирует, что матрица *A* будет полно- стью унимодулярной.

***Утверждение*** [13]***.*** *Матрица A, составленная из 0, 1 или —1, является полностью унимодулярной, если в каждом столбце не более двух ненулевых элементов, и строки матрицы A можно так разбить на два подмножества I*1 *и I*2*, что для каждого столбца верно следующее:*

1. *Если два ненулевых элемента имеют одинаковые знаки, то соответ- ствующие им строки лежат в разных множествах.*
2. *Если два ненулевых элемента имеют разные знаки, то соответствую- щие им строки лежат в одном и том же множестве*.

Рассмотрим задачу *LP* в следующем виде:

при ограничениях:

*n*

max *cjxj* (4.62)

∑

*j*=1

*n*

∑

*aijxj* = *bi, i* = 1*, . . . , m* (4.63)

*j*=1

*xj* ≥ 0*, j* = 1*, . . . , n, m < n.* (4.64)

Каким условиям должна удовлетворять матрица *A* = (*aij*) и вектор правых частей *b* = (*b*1*, . . . , bm*)*T* , чтобы все вершины многогранника (4.63)–(4.64) были целочисленными? Ответ на этот вопрос дает следующая теорема.

***Теорема*** [30]***.*** *Пусть целочисленная матрица A* = (*aij*)*, i* = 1*, . . . , m, j* = 1*, . . . , n, m < n имеет ранг m и вектор правых частей b* = (*b*1*, . . . , bm*)*T*

*— целочисленный. Для того, чтобы вершины многогранника* (4.63)–(4.64) *ле-*

*жали в целочисленных точках, необходимо и достаточно, чтобы матрица A*

*была полностью унимодулярной. Пример. Транспортная задача*

Пусть имеется *m* пунктов производства и *n* пунктов потребления однород- ного продукта. Объем производства в пункте *i* равен *ai*; объем потребления в

75

пункте *j* равен *bj*, *i* = 1*, . . . , m*, *j* = 1*, . . . , n.* Предполагается, что производство и потребление сбалансированы, т. е.

*m n*

∑ *ai* = ∑ *bj.* (4.65)

*i*=1

*j*=1

*i*=1

*j*=1

Задана матрица *c* = (*cij*) транспортных расходов на перевозку единицы про- дукции из пункта производства *i* в пункт потребления *j*. Требуется составить план перевозок, который, не нарушая производственные мощности, удовлетво- ряет все потребности с минимальными суммарными затратами на перевозки.

Введем переменные *xij* 0, обозначающие объем перевозок из пункта про- изводства *i* в пункт потребления *j*. Тогда получаем следующую задачу линей- ного программирования:

≥

∑ ∑

*m n*

при ограничениях:

min *cijxij* (4.66)

*i*=1 *j*=1

*n*

∑

*xij* = *ai, i* = 1*, . . . , m,* (4.67)

*j*=1

*m*

∑

*xij* = *bj, j* = 1*, . . . , n,* (4.68)

*i*=1

*xij* ≥ 0*, i* = 1*, . . . , m, j* = 1*, . . . , n.* (4.69)

Условие сбалансированности производства и потребления является необхо- димым и достаточным условием разрешимости транспортной задачи [13]. Чис- ло переменных *xij* в этой задаче равно *mn*, число ограничений равно *m* + *n*. Из-за условия сбалансированности производства и потребления строки (4.67) и (4.68) являются линейно зависимыми, следовательно, ранг системы ограниче- ний равен (*m* + *n* 1), а значит число ненулевых значений *xij* не превосходит (*m* + *n*)*.*

−

Если объемы производства *ai* и объемы потребления *bj* — целые числа, то любая вершина многогранника (4.67)–(4.69) целочисленная, поскольку матрица ограничений является полностью унимодулярной [4].

*Пример*

Пусть в модели среди всех ограничений имеется одно, которое нарушает унимодулярность всей матрицы. Рассмотрим вариант модификации этого огра- ничения в модели, который приводит к полностью унимодулярной подматрице. Пусть в модели имеются булевы переменные *xi*, *i* = 1*, . . . , n* и задача требует,

чтобы выполнялось следующее условие:

если *xn* = 0*,* то *x*1 = 0*, x*2 = 0*, . . . , xn*−1 = 0*.* (4.70)

76

Тогда неравенство

*x*1 + *x*2 + *. . .* + *xn*−1 ≤ (*n* − 1)*xn* (4.71)

гарантирует выполнение этого условия. Нетрудно заметить, что условие (4.70) можно смоделировать и по-другому, системой из (*n* − 1) неравенств:

*x*1 ≤ *xn, x*2 ≤ *xn,*

. (4.72)

*xn*−1 ≤ *xn.*

Построив для модели с представлением (4.72) двойственную задачу [14], нетруд- но заметить, что матрица ограничений соответствующей двойственной задачи будет полностью унимодулярной. Следовательно, решив линейную релаксацию двойственной задачи, можно найти оптимальное целочисленное решение исход- ной задачи, пользуясь средствами линейного программирования. Таким обра- зом, замена одного ограничения (4.71) системой ограничений (4.72) улучшает модель.

## Уточнение значения границ переменных

Рассмотрим распространенное в моделях *M IP* ограничение вида *y ux*, где *u* — некоторое положительное число, *x* 0*,* 1 . Насколько важно значение *u* с точки зрения разрыва целочисленности?

∈ { }

≤

Если переменная *x* принимает целые значения, то неважно, чему равно *u*, но если 0 *x* 1, то значение *u* оказывается существенным.

≤ ≤

Предположим, что максимальное допустимое значение, которое может при- нимать переменная *y*, равно *u*′ и *u*′ *< u*, а в целевой функции имеются слага- емые вида *f x*, соответствующие, например, затратам на запуск производства, *f >* 0. Если в оптимальном решении *y* = *u*′, то, учитывая ограничение *y* ≤ *ux*,

получаем, что *x* = *u′* и *x <* 1. Следовательно, затраты составят *f u′* . С другой

стороны, если бы ограничение было записано как *y u*′*x*, то при *y* = *u*′ полу- чили бы *x* = 1 и затраты составили бы величину *f* . Следовательно, чем грубее значение *u* по сравнению с *u*′, тем больше разрыв целочисленности.

*u*

*u*

≤

Иногда хорошие граничные условия можно определить аналитически.

*Пример*

В задаче о планировании производства, рассмотренной в разд. 4.2, в огра-

ничениях (4.28) для каждого промежутка времени *t* вместо грубой оценки

∑

*T*

*k*=1

*dk* лучше использовать уточненную оценку ∑*T*

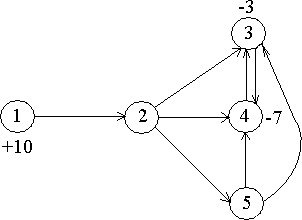
*dk*.

*Пример. Задача о потоке с фиксированными затратами*

*k*=*t*

Рассмотрим на примере задачи о потоке с фиксированными затратами ана- литические способы уточнения границ значений переменных. Пусть имеется сеть, как показано на рис. 23.

77



*Рис. 23.* Сеть для задачи о потоке с фиксированными затратами

Известны *cij* — удельные стоимости перевозки продукции вдоль дуги (*i, j*), *hij* — плата за использование дуги (*i, j*), *bi* — спрос или предложение в вер- шине *i*. Требуется доставить 3 и 7 единиц продукции в 3-ю и 4-ю вершины соответственно с минимальными затратами.

Введем неотрицательные переменные *xij* — величина потока, пропускаемого по дуге (*i, j*);

{*y* =*ij*

1*,* если по дуге (*i, j*) выполняются перевозки,

0 − в противном случае.

Пусть *B* — некоторое положительное большое число.

Математическая модель выглядит следующим образом:

min ∑ (*cijxij* + *hijyij*)*,* (4.73)

при ограничениях:

(*i,j*)∈*A*

*xij* ≤ *Byij,* (*i, j*) ∈ *A,* (4.74)

∑ *xji* − ∑ *xij* = *bi, i* ∈ *V,* (4.75)

*j*∈*V j*∈*V*

*xij* ≥ 0*,* (*i, j*) ∈ *A,* (4.76)

*yij* ∈ {0*,* 1}*,* (*i, j*) ∈ *A.* (4.77)

В данной задаче отсутствуют ограничения на пропускные способности дуг, поэтому в ограничениях (4.74) присутствует некоторое большое число *B*, что является недостатком данной модели. Уточним верхние оценки на принимае- мые значения для переменных *xij*, проанализировав веса вершин сети.

Итак, вес первой вершины составляет 10 единиц. Если дуга (1*,* 2) использу- ется, то по дуге *x*12 можно отправить не более 10 единиц, следовательно,

*x*12 ≤ 10*y*12*.*

78

Вторая вершина — транзитная, и доставить в нее можно не более 10 единиц, следовательно, и вывезти из нее по каждой выходящей дуге можно не более 10 единиц, значит

*x*23 ≤ 10*y*23*,*

*x*24 ≤ 10*y*24*, x*25 ≤ 10*y*25*.*

В третьей вершине потребляется 3 единицы продукции, поэтому если доставить максимально возможное количество продукции в размере 10 единиц в третью вершину, то вывезти из нее можно будет не более 7 единиц:

*x*34 ≤ 7*y*34*.*

Аналогично получаем следующие уточненные границы на значения перемен- ных:

*x*43 ≤ 3*y*43*,*

*x*54 ≤ 10*y*54*, x*53 ≤ 10*y*53*.*

Добавление новых неравенств вместо ограничений (4.74) позволяет сузить до- пустимую область и сократить разрыв.

## Удаление избыточных ограничений

Рассмотрим еще один способ предварительной обработки математической модели с булевыми переменными. Пусть допустимое множество задается груп- пой линейных неравенств вида

*n*

∑

*ajxj* ≤ *b*

*j*=1

*xj* ∈ {0*,* 1}*, j* = 1*, . . . , n.*

Если *aj <* 0 для некоторого *j*, тогда заменой переменной *xj* на (1 *x*′*j* ) можно получить эквивалентное ограничение:

−

*j*=1*,.*∑*..,n*|*aj >*0

*ajxj* +

∑

*j*=1*,...,n*|*aj <*0

|*aj*|*x*′*j* ≤ *b* −

*j*=1*,...,n*|*aj <*0

∑

*aj.*

Это сведение позволяет без ограничения общности считать, что все *aj >* 0, *j* = 1*, . . . , n*. Если *j N aj > b*, для некоторого *N* 1*, . . . , n* , то в модель можно ввести дополнительное неравенство

∑ ⊆ { }∈

∑

*xj* ≤ |*N* | − 1 (4.78)

*j*∈*N*

79

или несколько неравенств для каждого *k N* , при котором *j N k aj b*.

∑∈ ≤∈ \{ }

Возможно, что после таких преобразований удастся объединить некоторые

неравенства и зафиксировать значения некоторых переменных, сократить раз- мерность задачи.

*Пример*

Рассмотрим следующую систему неравенств:

−3*x*2 − 2*x*3 ≤ −2*,* (4.79)

−4*x*1 − 3*x*2 − 3*x*3 ≤ −6*,* (4.80)

2*x*1 − 2*x*2 + 6*x*3 ≤ 5*,* (4.81)

*xi* ∈ {0*,* 1}*, i* = 1*,* 2*,* 3*.* (4.82)

Выполним замену переменных, получим:

3*x*′2 + 2*x*′3 ≤ 3*,* (4.83)

4*x*′1 + 3*x*′2 + 3*x*′3 ≤ 4*,* (4.84)

2*x*1 + 2*x*′2 + 6*x*3 ≤ 7*,* (4.85)

*x*1*, x*′*i* ∈ {0*,* 1}*, i* = 1*,* 2*,* 3*.* (4.86)

Учитывая (4.78) для (4.83), получаем

*x*′2 + *x*′3 ≤ 1 или *x*2 + *x*3 ≥ 1*,* (4.87)

для (4.85) получаем

*x*′2 + *x*3 ≤ 1 или *x*3 ≤ *x*2*.* (4.88)

Объединяя (4.87) и (4.88), получаем *x*2 = 1 или *x*′2 = 0, и тогда ограничение (4.79) избыточное, а ограничения (4.84) и (4.85) превращаются в

4*x*′1 + 3*x*′3 ≤ 4 и 2*x*1 + 6*x*3 ≤ 7*.*

Учитывая (4.78), получаем

*x*′1 + *x*′3 ≤ 1 или *x*1 + *x*3 ≥ 1

и

*x*1 + *x*3 ≤ 1*,*

следовательно, *x*1 + *x*3 = 1. Используя это равенство, можно исключить одну из переменных *x*1 или *x*3. Таким образом, удается сократить число переменных и ограничений.

На этом изложение идей и фактов о том, как строить модели, оценивать их качество и выбирать среди возможных моделей наилучшую, завершается. Заинтересованный читатель может детально познакомиться с предложенными техниками из литературы, приведенной в конце пособия. В следующем разделе предлагаются упражнения, направленные на усвоение прочитанного материа- ла.

80

# Упражнения

В этом разделе собраны упражнения на моделирование, поиск наилучшего представления допустимого множества и построение правильных неравенств. Упражнения разделены на две части. В первой части собраны задачи, которые решаются без помощи специальных программных средств. Во второй части приводятся задания, для выполнения которых сначала рекомендуется постро- ить математическую модель, а затем найти решение с помощью программных средств. Существует много коммерческих и бесплатных программных обеспече- ний для решения оптимизационных задач [16 — 20]. Читатель может выбрать любой из них, например, систему GAMS. Размерность задач в упражнениях подобрана таким образом, что можно построить математическую модель, ко- личество ограничений и целочисленных переменных которой будет не слишком велико. Для тех, кто незнаком с этой системой, в следующем разделе разобран пример решения задачи в GAMS.

## Теоретические задания

1. Пусть *xi* 0*,* 1 *, i* 1*, . . . , n* . Что можно сказать о значениях каждой из переменных, удовлетворяющих следующим неравенствам:

∈ { } ∈ { }

* 1. *xi* + *xj* ≤ 1 и *xi* ≤ *xj*, для некоторых *i, j* ∈ {1*, . . . , n*}*,*
  2. *xi* + *xj* ≤ 1 и *xi* + *xj* ≥ 1*,* для некоторых *i, j* ∈ {1*, . . . , n*}*,*
  3. *xi* ≤ *xj* и *xj* + *xk* ≤ 1*,* для некоторых *i, j, k* ∈ {1*, . . . , n*}?

1. Пусть область *S* задается следующим образом:

∑ *ajxj* − ∑ *ajxj* ≤ *b,*

*j*∈*N*1

*j*∈*N*2

*j*∈*N*1

*j*∈*N*2

*xj* ∈ {0*,* 1}*, j* ∈ {*N*1} ∪ {*N*2}*,*

где *aj >* 0. Выпишите необходимые и достаточные условия, чтобы 1) *S* = ∅*,*

* 1. *S* совпадало со всеми 0–1 векторами размерности |*N*1| + |*N*2|,
  2. *xj* = 0 для всех *x* ∈ *S,*
  3. *xj* = 1 для всех *x* ∈ *S,*
  4. *xi* + *xj* ≤ 1 для всех *x* ∈ *S,*
  5. *xi* ≤ *xj* для всех *x* ∈ *S,*
  6. *xi* + *xj* ≥ 1 для всех *x* ∈ *S.*

1. Представьте произведение *y*1*y*2*y*3 трех (четырех и т.д.) булевых перемен- ных линейными ограничениями.

81

1. Пусть *xj* 0*, j.* Задайте выполнение по крайней мере одного из двух неравенств

≥ ∀

∑ *a*1*jxj* ≤ *b*1 или ∑ *a*2*jxj* ≤ *b*2*,*

*j*

*j*

используя только линейные ограничения и одну вспомогательную булеву переменную.

1. Пусть *xj* ≥ 0*,* ∀*j.* Задайте выполнение одного из двух неравенств

∑ *a*1*jxj* ≤ *b*1 или ∑ *a*2*jxj* ≤ *b*2*,*

*j*

*j*

*j*

*j*

используя только линейные ограничения и одну вспомогательную булеву переменную.

1. Задайте выполнение одного из *n* неравенств

∑

*a*1*jxj* ≤ *b*1

*j*

или

*a*2*jxj* ≤ *b*2*,*

∑

*j*

или

.

или

*anjxj* ≤ *bn,*

∑

*j*

используя только линейные ограничения и *n* вспомогательных булевых переменных.

1. Пусть кусочно-линейная функция *f* (*x*) задана следующим образом:

*f* (*x*) =

*c*0(*x* − *a*0)*,* если *a*0 *< x* ≤ *a*1*,*

*c*0*a*1 + *c*1(*x* − *a*1)*,* если *a*1 *< x* ≤ *a*2*, c*0*a*1 + *c*1(*a*2 − *a*1) + *c*2(*x* − *a*2)*,* если *a*2 *< x* ≤ *a*3*,*



 0*,* если *x* = *a*0*,*

где *c*0 = 50, *c*1 = 207, *a*0 = 0, *a*1 = 140, *a*2 = 220, *a*3 = 420. Постройте MIP поиска минимума *f* (*x*), *x* ∈ [*a*0*, a*3].

1. Смоделируйте с помощью линейный ограничений, чтобы переменная *y*

принимала значение, равное max(*u*1*, u*2), где *u*1*, u*2 ≥ 0.

82

1. Совет директоров должен принять решение о выборе инвестиций. Име- ется 7 вариантов вложений. На множестве инвестиций заданы условия вложений (табл. 6).

*Таблица 6*

Условия инвестиций

|  |  |  |
| --- | --- | --- |
| Тип | инвестиции | Условия вложений |
| 1 | | — |
| 2 | | если инвестировали в 1 |
| 3 | | если инвестировали в 2 |
| 4 | | если инвестировали и в 1, и в 2 |
| 5 | | если не было инвестиций в 1 или в 2 |
| 6 | | если не было инвестиций ни в 2, ни в 3 |
| 7 | | если были инвестиции в 2, и не было в 3 |

Известны *ri* — доход от инвестиции *i*, *ci* — расходы при вложении *i*. Объем инвестиций не должен превышать *M* y.e. Компания стремится получить максимальную прибыль от вложений. Постройте математическую модель.

1. Компания должна назначить 8 команд на 3 проекта: *A*, *B* и *C*. Доход от каждого проекта зависит от числа команд, назначенных на проект. Ни один проект не должен остаться без команды. Выполнять один проект может не более 5 команд. Каждая команда может быть назначена толь- ко на один проект. Доход от соответствующих назначений приводится в табл. 7.

*Таблица 7*

Доход

|  |  |
| --- | --- |
| Количество команд | Проект  *A B C* |
| одна | 45 20 50 |
| две | 70 45 70 |
| три | 90 75 80 |
| четыре | 105 110 100 |
| пять | 120 150 130 |

Цель компании — назначить команды на проекты так, чтобы максимизи- ровать суммарный доход.

Ответьте на следующие вопросы.

**а.** Сколько ограничений нужно включить в модель:

* 1. одно;
  2. два или три;

83

* 1. четыре или пять;
  2. по крайней мере шесть?

**б.** Какое из следующих утверждений неверно:

1. все коэффициенты при переменных во всех ограничениях либо 0, либо 1, либо −1;
2. в модели все переменные — булевы;
3. можно записать модель, используя ограничения только в виде ра- венств;
4. в оптимальном решении на проекты назначены все 8 команд?

**в.** Предположим, что появляется еще один проект. Сколько потребуется новых переменных:

1) 0;

2) 1;

3) 3;

4) 5?

**г.** Вернитесь к исходной задаче. Предположим, что появляется еще одна команда. Сколько потребуется новых переменных:

1) 0;

2) 1;

3) 3;

4) 5?

1. Для обслуживания *m* клиентов могут быть задействованы *n* грузовиков. Каждого клиента можно посетить только один раз. Время на дорогу от клиента *i* до клиента *j*, которое затрачивает грузовик *k*, равно *cijk*, *i, j* 1*, . . . , m* , *k* 1*, . . . , n* . Общее время, в течение которого может быть задействован грузовик *k*, не должно превышать величину *lk*. Постройте модель целочисленного линейного программирования допустимой схемы объезда всех клиентов.

{ } ∈ { }

∈

1. Для обслуживания склада используется два грузовика *A* и *B*. Каждый грузовик может выезжать в 13:00, 14:00, 15:00 или в 16:00. Грузовик *B* не должен выезжать в течение часа после выезда грузовика *A*. Например, если грузовик *A* выехал в 13:00, то грузовик *B* может выезжать в 15:00 или в 16:00. Каждый грузовик выезжает со склада в течение дня один раз.

84

Используя переменные

{*x* =*i*

1*,* если грузовик *A* выехал в момент времени *i,*

0 − в противном случае,

{*y* =*i*

1*,* если грузовик *B* выехал в момент времени *i,*

0 − в противном случае,

предложите два разных способа построения множества допустимых вы- ездов грузовиков и сравните их.

1. Компания занимается доставкой посылок. Нужно развести посылки 10 клиентам. Заданы величины *dj* — количество посылок, которые нужно доставить клиенту *j*, *j* = 1*, . . . ,* 10. В компании имеется 4 грузовика вме- стимостью *qi* посылок каждый, *i* = 1*, . . . ,* 4. Предполагается, что все по-

*j*=1

сылки не входят в один грузовик, но ∑4

*i*=1

*qi* ≥ ∑10

*dj* и *dj* ≤ *qi*. Партию

посылок предназначенную одному клиенту нельзя доставлять разными

грузовиками. Затраты при отправке грузовика *i* равны *ci*, *i* = 1*, . . . ,* 4. Каждый грузовик может объехать не более пяти клиентов, кроме того, к следующим парам клиентов нельзя доставлять посылки одним и тем же грузовиком: (1,7), (2,6), (2,9). Определите, какие грузовики необходимо задействовать, чтобы доставить все посылки клиентам с минимальными суммарными затратами. Постройте математическую модель.

1. Отдел статистики подсчитал, что в городе имеется *n* женихов и *n* невест. Известна привлекательность (*cij*) невесты *i* женихом *j*. Постройте ма- тематическую модель так, чтобы каждому жениху назначить невесту и каждой невесте назначить жениха с максимальной суммарной привлека- тельностью между парами. Правда ли, что задача является полиноми- ально разрешимой?
2. Компания производит два типа продукции на одном заводе, чтобы удо- влетворить запросы клиентов в пяти регионах. Поставка продукции осу- ществляется через два крупных центра. Величина *djk* определяет запрос клиентов из региона *j* на продукт *k*, *k* = 1*,* 2, *j* = 1*, . . . ,* 5. Компания должна решить, какой тип продукта в какой центр доставить и как его распределить по регионам, чтобы минимизировать суммарные затраты. Затраты складываются из:

фиксированных затрат *fik*, возникающих, если продукт типа *k* по- ставляется из центра *i*;

•

фиксированных затрат *gijk*, возникающих, если продукт типа *k* по- ставляется доставляется клиенту в регион *j* через центр *i*;

•

удельных затрат *cijk* на транспортировку продукта типа *k* в регион

•

*j* через центр *i*.

85

Постройте математическую модель.

1. У авиакомпании имеется расписание вылета воздушных судов в пять горо- дов на ближайший день. Для обслуживания самолетов необходимо нанять экипажи и составить для них расписание. Время работы каждого экипажа в течение суток ограничено. Необходимо определить, какое минимальное количество экипажей нужно нанять и составить для них расписание, что- бы все запланированные вылеты состоялись. Постройте математическую модель, используя задачу о покрытии.
2. Имеется *n* работ и одна машина для выполнения этих работ. В каждый момент времени может выполняться только одна работа. Каждая работа выполняется без прерывания. Известны следующие целочисленные вели- чины:

*pj* — длительность выполнения работы *j*,

*dj* — крайний срок завершения каждой работы и *wj* — важность работы,

*j* = 1*, . . . , n*.

* 1. Требуется найти допустимое расписание, минимизирующее взвешен- ную сумму времен завершения работ. Постройте математическую мо- дель. Избегайте в модели введения вспомогательных больших чисел, используйте точные оценки на время окончания работ.
  2. Постройте модель, используя следующие переменные:

 1*,* если работа *j* завершилась в момент

*xjt* =

времени *t,*

 0 − в противном случае.

1. Компания имеет две машины для производства пластиковых бутылок. В течение года нужно проводить техническое обслуживание оборудова- ния. Техническое обслуживание каждой машины длится 2 месяца под- ряд. Кроме того, в декабре и январе половина рабочих отдыхает, по- этому в производственном процессе может быть задействована только одна машина, а другая машина в это время может проходить техниче- ское обслуживание. Ежемесячный спрос на бутылки составляет *dt* штук, *t* = 1*, . . . ,* 12. Максимальное количество бутылок, которые машина *k* мо- жет произвести в месяц составляет *ak* штук, *k* = 1*,* 2. Для того, чтобы произвести *ak* бутылок, машина должна проработать *lk* рабочих дней. В каждом месяце только *Lt* рабочих дней, *t* = 1*, . . . ,* 12.

Необходимо составить допустимый график проведения технического об- служивания машин при условии, что спрос на бутылки будет удовлетво- рен. Постройте математическую модель.

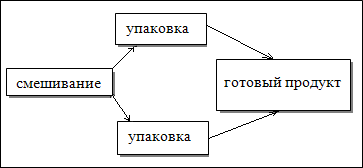
Как изменится модель, если требуется:

* 1. минимизировать суммарные ежемесячные скачки в загруженности ра- бочих дней;

86

* 1. минимизировать наибольший скачок в течение года между загружен- ными рабочими днями.

1. Завод производит *N* разных продуктов из злаков и фруктов. Производ- ственный процесс состоит из трех этапов: подготовка сырья (очистка, сушка и т. д.), смешивание (расфасовка) и упаковка. На рис. 24 изобра- жена общая схема работы завода. Сырье поставляется в неограниченном



*Рис. 24.* Схема работы завода

количестве. Для смешивания и расфасовки фруктов и злаков использу- ется одна машина. После этого готовый продукт упаковывается на одной из двух машин.

Время работы каждой машины *k* ограничено величиной *Lk*, производи- тельность машины *k* равна *αik* единиц продукта *i* в час, *i* = 1*, . . . , N* , *k* = 1*,* 2*,* 3.

После смешивания и расфасовки продукта *i* машину необходимо промыть в течение *βi* часов. Завод имеет склад, и первоначально на складе имеется *Si* единиц упакованного продукта *i*, а к концу каждой недели *t* на складе должно оставаться *Si* единиц упакованного продукта *i* в качестве резерва,

0

*t*

*i* = 1*, . . . , N* .

Необходимо спланировать производство продуктов на *T* недель вперед так, чтобы на складе оставалось как можно меньше готовой продукции за весь период планирования, при этом удовлетворить спрос *Di* единиц упакованного продукта *i* на неделю *t*. Постройте математическую модель.

*t*

1. Вспомните два представления допустимого множества решений задачи коммивояжера (разд. 3.2). Какое из этих представлений лучше?
2. Вспомните задачу коммивояжера, предложенную в разд. 3.2, и предло-

87

жите представление, используя следующие переменные:

{*x* =*ijk*

1*,* если дуга (*i, j*) идет *k*-й по порядку в обходе,

0 − в противном случае.

1. Вспомните двухстадийную задачу о гильотинном раскрое, рассмотренную в разд. 3. Постройте математическую модель при условии, что прямо- угольники можно поворачивать на 900.
2. Вспомните задачу о башнях, рассмотренную в разд. 3. В том примере пять из шести возможных вариантов размещения башен приводят к разбросу клиентов, равному двум, и только один вариант с открытием 2-й и 3-й башен дает равномерное распределение клиентов с перепадом, равным нулю. Придумайте пример, т. е. подберите соответствующие значения *n*, *m* и *p*, и задайте матрицу (*cij*), демонстрирующую, что перепад между максимальным и минимальным количеством клиентов может быть боль- шим.
3. Задан взвешенный неориентированный граф *tt* = (*V, E*) со множеством вершин *V* и множеством ребер *E*. Каждому ребру *e E* приписан неот- рицательный вес *we*. Под весом разреза для произвольного разбиения множества вершин на непересекающиеся подмножества будем понимать сумму весов ребер, каждое из которых имеет вершины в разных подмно- жествах. Для заданного натурального числа *p* задача заключается в том, чтобы найти разбиение множества вершин *V* на подмножества мощности не более *p*, имеющее минимальный вес разреза.

∈

Представьте задачу в терминах целочисленного линейного программиро- вания. Имеет ли полученная математическая модель симметрии? Если да, то удалите симметрии введением дополнительных ограничений.

1. Любитель фильмов обнаружил в компьютерной сети новую библиотеку с фильмами и решает просмотреть все фильмы. Он знает, сколько нуж- но времени для того, чтобы скачать, просмотреть один фильм, и размер каждого фильма. Размер доступного пространства для хранения видео- файлов на диске его компьютера ограничен. Поэтому любитель сначала полностью скачивает фильм, затем просматривает его и удаляет с дис- ка. Скачивание и просмотр видео происходит без прерываний. Пока один фильм просматривается, другой фильм может закачиваться на диск. Ска- чивание очередного фильма начинается, только если перед началом ска- чивания он может целиком поместиться на диск.

Требуется найти порядок скачивания и просмотра фильмов, чтобы закон- чить просмотр всех фильмов как можно быстрее.

Постройте математическую модель целочисленного программирования. Правда ли, что среди оптимальных решений задачи всегда найдется ре- шение с одинаковым порядком скачивания и просмотра фильмов?

88

1. Заданы матрица (*hij*), *i* = 1*, . . . , n*, *j* = 1*, . . . , m* с положительными ве- щественными элементами и вектор (*cj*), *j* = 1*, . . . , m* с положительными вещественными элементами. Пусть F — семейство всех непустых подмно- жеств множества {1*, . . . , m*}. Для каждого *F* ∈ F определим

*z*(*F* ) = ∑ max *hij* − ∑ *cj.*

*j*∈*F*

*j*∈*F*

*n*

*i*=1 *j*∈*F*

Постройте для задачи максимизации функции *z*(*F* ) модель целочисленно- го линейного программирования, используя только булевы переменные.

1. Докажите, что если все компоненты матрицы *A* и вектор-столбца *b* це- лочисленные и *A* — унимодулярная матрица, то все базисные решения задачи

min *cx Ax* = *b x* ≥ 0

имеют целочисленные компоненты. Докажите, что утверждение верно, если *Ax* ≤ *b* (*Ax* ≥ *b*).

1. Множество *S* может быть задано несколькими способами:

97*x*1 + 32*x*2 + 25*x*3 + 20*x*4 ≤ 139*, xi* ∈ {0*,* 1}*, i* = 1*, . . . ,* 4

или

или

2*x*1 + *x*2 + *x*3 + *x*4 ≤ 3*, xi* ∈ {0*,* 1}*, i* = 1*, . . . ,* 4

*x*1 + *x*2 + *x*3 ≤ 2*, x*1 + *x*3 + *x*4 ≤ 2*, x*1 + *x*2 + *x*4 ≤ 2*,*

*xi* ∈ {0*,* 1}*, i* = 1*, . . . ,* 4*.*

Какой из способов наиболее эффективный при решении задачи max*x*∈*S cx*?

1. Пользуясь результатами из упражнений 1 и 2, решите следующую задачу, не делая перебора:

при условиях:

max 2*x*1 − 2*x*2 + 3*x*3 + *x*4

7*x*1 + 3*x*2 + 9*x*3 − 2*x*4 + 2*x*5 ≤ 7*,*

−6*x*1 + 2*x*2 − 3*x*3 + 4*x*4 + 9*x*5 ≤ −2*, xj* ∈ {0*,* 1}*, j* = 1*, . . . ,* 5*.*

89

1. Пусть множество *S* задается следующей системой неравенств:

*x*1 − *x*2 ≤ 1*,*

4*x*1 + *x*2 ≤ 28*, x*1 + 4*x*2 ≤ 27*,*

*x*1 ≥ 0*, x*2 ≥ 0*,* целые.

Найдите систему неравенств, задающую выпуклую оболочку множества

*S*.

1. Многогранник *P* задается следующей системой неравенств:

*x*1 − *x*2 ≤ 0*,*

−*x*1 + *x*2 ≤ 1*,* 2*x*2 ≥ 5*,*

8*x*1 − *x*2 ≤ 16*, x*1 + *x*2 ≥ 4*,*

*x*1 ≥ 0*, x*2 ≥ 0*,* целые.

Найдите «наименьшее» представление для многогранника *P* .

1. Найдите *conv*(*X*), если множество *X* задано следующим образом:

−*x*1 + 2*x*2 ≤ 7*,*

*x*1 + 3*x*2 ≤ 15*,* 7*x*1 − 3*x*2 ≤ 23*,*

*x*1 ≥ 0*, x*2 ≥ 0*,* целые.

1. Найдите наилучшее представление для множества *X*, заданного следую- щим образом:

−*x*1 + *x*2 ≤ 3*, x*2 ≤ 4*,*

7*x*1 + 11*x*2 ≤ 65*,*

7*x*1 − 3*x*2 ≤ 23*,*

*x*1 ≥ 0*, x*2 ≥ 0*,* целые.

90

1. Пусть многогранник задается следующей системой неравенств:

*x*1 + *x*2 ≤ *b*1*,*

−*x*1 + *x*2 ≤ *b*2*, x*1 ≥ 0*, x*2 ≥ 0*,*

где *b* = (*b*1*, b*2)*T* — целочисленный вектор с неотрицательными компонен- тами.

Ответьте на следующие вопросы.

1. Является ли матрица ограничений полностью унимодулярной?
2. Существует ли вектор правых частей *b*, такой что все вершины много- гранника будут целочисленными точками?
3. Рассмотрим допустимую область задачи, известной в литературе как за- дача размещения с ограничениями на мощности производства:

∑

*yij* = *ai* для каждого *i* ∈ *I,* (5.1)

*j*∈*J*

*yij* ≤ *bjxj* для каждого*j* ∈ *J,* (5.2)

∑

*i*∈*I*

*yij* ≥ 0*, xj* ∈ {0*,* 1} для каждого *i* ∈ *I, j* ∈ *J.* (5.3) Пусть *I*′ ⊆ *I* и *zj* = ∑*i*∈*I′ yij*, такие, что

∑ *zj* = ∑ *ai* и *zj* ≤ *bjxj.*

*j*∈*J*

*i*∈*I′*

*j*∈*J*

*i*∈*I′*

Постройте правильные неравенства для (5.1)–(5.3).

1. Рассмотрим множество:

*n*

∑*X* = {*y* ∈ : *C y* ≥ *b*}*,* (5.4)*j*

Z*n*

+

*j*=1

где *C*, *b* — целые числа.

1. Найдите *conv*(*X*).
2. Найдите *conv*(*X* ∩ {0*,* 1}*n*).
3. Рассмотрим задачу:

min −*x*1 − *x*2 − *x*3 *x*1 + *x*2 ≤ 1

91

*x*2 + *x*3 ≤ 1

*x*1 + *x*3 ≤ 1

*x*1*, x*2*, x*3 ≥ 0*,* целые.

1. Найдите разрыв целочисленности.
2. Постройте правильное неравенство.

## Практические задания

1. Ресторан работает 7 дней в неделю. Повара работают 6 часов в день, 5 дней подряд и затем 2 дня отдыхают. У всех поваров одинаковая зарплата. Приготовление каждого блюда занимает определенное время, и для каж- дого дня недели установлено общее необходимое количество часов для приготовления пищи. Данные приведены в табл. 8.

Администратору нужно решить, какое количество поваров нанять и в какие дни они должны работать, чтобы нужное количество часов было отработано, а затраты на оплату труда были минимальными.

*Таблица 8*

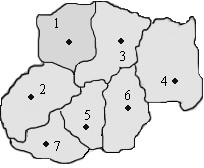
Ежедневное количество рабочих часов

|  |  |
| --- | --- |
| День недели | Требуемое количество часов |
| понедельник | 150 |
| вторник | 200 |
| среда | 400 |
| четверг | 300 |
| пятница | 700 |
| суббота | 800 |
| воскресенье | 300 |

Постройте математическую модель. Найдите оптимальное решение.

1. Авиакомпания «Альфа» составляет расписание вылетов из Чикаго по сле- дующим направлениям: Колумбия, Денвер, Лос-Анджелес и Нью-Йорк. В каждый город должен состояться ровно один вылет. Вылеты могут быть в 8:00, 10:00 и 12:00. Авиакомпания оплачивает вылет каждого самолета по каждому направлению. Эти затраты составляют 5 тыс. у.е., если вылет совершается до 10:00 включительно, и 3 тыс. у.е. — после 10:00. В каж- дый момент времени выполняется не более двух рейсов. Кроме того, если в определенное время есть вылет в Нью-Йорк, то в это же время должен быть вылет в Лос-Анджелес. Ожидаемый доход (в тыс. у.е.) от полетов приводится в следующей табл. 9.

92



*Рис. 25.* Расположение районов города

*Таблица 9*

Ожидаемый доход от полетов

|  |  |  |  |
| --- | --- | --- | --- |
| Направление Время вылета  8:00 10:00 12:00 | | | |
| Колумбия | 10 | 6 | 6 |
| Денвер | 9 | 10 | 9 |
| Лос-Анджелес | 14 | 11 | 10 |
| Нью-Йорк | 18 | 15 | 10 |

Требуется составить расписание, доставляющее максимальную прибыль авиакомпании. Постройте математическую модель. Найдите оптимальное решение.

1. В городе 7 районов (рис. 25). Известна численность населения в каждом районе (табл. 10).

*Таблица 10*

Численность населения

|  |  |
| --- | --- |
| Район | Численность населения(тыс. человек) |
| 1 | 16 |
| 2 | 15 |
| 3 | 98 |
| 4 | 10 |
| 5 | 4 |
| 6 | 6 |
| 7 | 100 |

93

В институте по повышению уровня грамотности работает 2 специалиста. В какие районы нужно отправить специалистов, чтобы уровень грамот- ности в городе стал как можно выше? Если специалист находится в одном из районов, он может посетить еще один соседний район. Районы называ- ются соседними, если их границы на карте города — смежные. Постройте математическую модель. Найдите оптимальное решение.

1. Издательство планирует выпустить новую серию книг по исследованию операций. В серию войдут три книги: *OR*1, *OR*2, *OR*3. Авторы сдают в редакцию рукопись книги, а издательство брошюрует и печатает книгу. В издательстве имеются три машины для печати книг — *P*1, *P*2, *P*3 и две машины для брошюровки — *B*1, *B*2. Время работы каждой машины с соответствующей книгой указано в табл. 11.

*Таблица 11*

Время работы машин

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *P*1 | | *P*2 | *P*3 | *B*1 | *B*2 |
| *OR*1 | 3 | 6 | 4 | 10 | 10 |
| *OR*2 | 2 | 3 | 3 | 12 | 11 |
| *OR*3 | 4 | 5 | 5 | 15 | 14 |

Машины могут работать только определенное количество часов. Макси- мальное время работы (в часах) каждой машины указано в табл. 12.

*Таблица 12*

Максимальное время работы машин

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *P*1 | | *P*2 | *P*3 | *B*1 | *B*2 |
| время работы | 120 | 100 | 110 | 333 1  3 | 300 |

Стоимость аренды одной машины не зависит от числа книг и составляет 10, 8, 9, 20 и 23 тыс. y.e. соответственно. Доход от продажи одной книги составляет 40, 60 и 70 y.e. соответственно. Издательство стремится полу- чить максимальную прибыль. При этом оно должно выпустить не менее 500 штук книг *OR*3, чтобы поддерживать хороший академический имидж. Постройте математическую модель. Найдите оптимальное решение.

1. Трое рабочих *W*1, *W*2, *W*3 должны выполнить пять работ *J*1, *J*2, *J*3, *J*4, *J*5. Уровень подготовки и опыт работы у рабочих разный. Время выполнения конкретной работы в часах у каждого работника приводится в табл. 13.

94

*Таблица 13*

Время выполнения работы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *J*1 | | *J*2 | *J*3 | *J*4 | *J*5 |
| *W*1 | 5 | 1 | 9 | 4 | 9 |
| *W*2 | 4 | 3 | 8 | 3 | 8 |
| *W*3 | 7 | 5 | 6 | 4 | 7 |

Каждая работа выполняется рабочим без прерывания. Требуется распре- делить и выполнить все работы так, чтобы время загрузки всех рабо- чих было равномерным. Предложите несколько вариантов моделирова- ния равномерной загруженности рабочих (с помощью линейной и квад- ратичной целевых функций). Найдите оптимальное решение для разных моделей.

1. В лесничестве нужно принять решение, какие участки леса будут выруб- лены под застройку. Лес разделен на 16 участков прямоугольной формы, как показано в табл. 14.

*Таблица 14*

Разделение леса на участки

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 |

Соседние участки нельзя выбирать под застройку, чтобы избежать боль- ших вырубленных площадей. Соседними для шестого участка являются, например, участки с номерами 2, 5, 7 и 10. Польза от застройки каждого участка приводится в табл. 15.

*Таблица 15*

Полезность застройки каждого участка

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 6 | 2 | 5 |
| 4 | 3 | 5 | 8 |
| 8 | 2 | 4 | 9 |
| 10 | 1 | 6 | 7 |

Определите, какие участки выбрать, чтобы общая польза от их застройки была максимальной. Постройте математическую модель, найдите опти- мальное решение задачи.

95

1. Компания «Гига» владеет тремя складами *S*1, *S*2, *S*3 вместимостью 30, 10 и 50 тыс. тонн соответственно. Фиксированные затраты на подготовку к использованию каждого склада составляют 25, 50 и 45 тыс. y.e. соответ- ственно. Первый склад можно расширить за счет использования подзем- ного хранилища на 20 тыс. тонн с дополнительными затратами 1 тыс. y.e. за тонну. При необходимости имеется возможность открыть еще два скла- да *S*4, *S*5 вместимостью 20 и 30 тыс. тонн с затратами на подготовку 15 и 25 тыс. y.e. соответственно. В настоящий момент в «Гигу» обратились три компании *K*1, *K*2, *K*3 для хранения 20, 60 и 40 тыс. тонн. Компания «Ги- га» забирает грузы у клиентов и привозит их на склады, транспортные затраты на доставку приводятся в табл. 16.

*Таблица 16*

Транспортные затраты на доставку

|  |  |  |  |
| --- | --- | --- | --- |
|  | *K*1 | *K*2 | *K*3 |
| *S*1 | 3 | 7 | 4 |
| *S*2 | 2 | 6 | 8 |
| *S*3 | 9 | 3 | 4 |
| *S*4 | 5 | 6 | 4 |
| *S*5 | 7 | 3 | 9 |

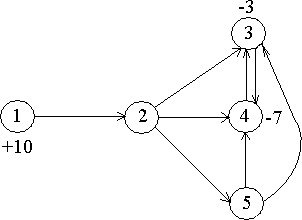
Требуется разместить грузы всех клиентов с минимальными суммарны- ми затратами. Нужно ли открывать дополнительные склады? Постройте математическую модель. Найдите оптимальное решение задачи.

1. Импресарио готовит выставку старинных автомобилей, среди которых могут быть Buggati, Cadillac, Cobra, Corvette, Pierce Arrow, Studebaker. Опрос показал, что посмотреть именно Buggati придут 58 специально приглашенных гостей, Cadillac — 37, Cobra — 42, Corvette — 40, Pierce Arrow — 55 и Studebaker — 33. Бюджет организации выставки составля- ет 15 млн y.e. Стоимость доставки автомобиля на выставку и обеспечение его сохранности составляют 6, 4, 3.8, 4.2, 5.5 и 3.2 млн y.e. соответственно. Задача импресарио в том, чтобы привлечь как можно больше специально приглашенных гостей, не превышая бюджет на организацию. Кроме то- го, на выставке должно быть не менее трех старинных автомобилей. Если Corvette будет выбран для выставки, то и Cobra должен там быть. Если же Buggati отсутствует, то обязательно нужно включить в показ Cadillac.

Постройте математическую модель и найдите оптимальное решение зада- чи. Определите, каким может быть минимальный и максимальный бюд- жет, чтобы выставка состоялась.

1. Компания распространяет технику по пяти городам: Екатеринбург, Омск, Новосибирск, Томск, Иркутск. В настоящий момент 10 снегоуборочных

96



*Рис. 26.* Сеть дорог

машин находятся в Екатеринбурге и должны быть доставлены в Ново- сибирск и Томск. В Новосибирск нужно 3 машины, в Томск — 7 машин. Пронумеруем города Екатеринбург, Омск, Новосибирск, Томск и Иркутск целыми числами от 1 до 5 соответственно. Сеть дорог между городами изображена на рис. 26. Вершины сети — города, дуги — дороги меж- ду городами. Некоторым вершинам предписан вес — положительное или отрицательное число. Положительное число означает, что в городе соот- ветствующему этой вершине, есть предложение продукции, равное весу вершины, отрицательный вес говорит о том, что в этой вершине имеет- ся спрос на продукцию, соответствующий весу вершины. Предполагается, что сумма весов всех вершин сети равна нулю, это означает, что суммар- ное предложение совпадает с суммарным спросом.

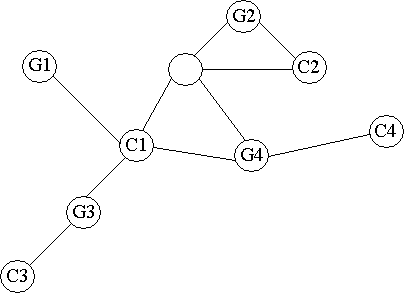
*Таблица 17*

Пропускные способности и стоимости перевозок

|  |  |  |
| --- | --- | --- |
| Дуга | Пропускная способность | Стоимость перевозки |
| (1,2) | 10 | 11 |
| (2,3) | 2 | 1 |
| (2,4) | 3 | 1 |
| (2,5) | 7 | 2 |
| (3,4) | 3 | 2 |
| (4,3) | 2 | 2 |
| (5,4) | 4 | 5 |
| (5,3) | 7 | 4 |

Количество водителей, осуществляющих перегон техники из одного горо- да в другой, ограничено, поэтому число снегоуборочных машин, которое

97



*Рис. 27.* Электроэнергетическая сеть

можно перевезти из одного города в другой, не должно превышать про- пускной способности дуги. Пропускные способности и стоимости перево- зок приводятся в табл. 17.

Необходимо составить план перевозок минимальной стоимости так, что- бы удовлетворить спрос и не нарушить пропускных возможностей при перегоне техники.

1. На рис. 27 изображена электроэнергетическая сеть, соединяющая элек- трогенераторы *tt*1, *. . .*, *tt*4 с источниками потребления *C*1, *. . .*, *C*4. Потоки энергии могут идти в любом направлении по ребрам сети, пропускные способности ребер неограничены, стоимость передачи по одному ребру составляет 11 y.e. за 1000 кВт/час. Мощность каждого электрогенерато- ра и стоимость выработки электроэнергии приводятся в табл. 18.

*Таблица 18*

Данные о выработке электроэнергии

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Электрогенератор  *tt*1 *tt*2 *tt*3 *tt*4 | | | |
| Мощность (тыс. кВт/час) | 100 | 60 | 80 | 150 |
| Стоимость выработки (за 1000 кВт/час) | 15 | 13,5 | 21 | 23,5 |

98

Энергопотребление в *C*1, *. . .*,*C*4 составляет 35, 50, 60 и 40 тыс. кВт соот- ветственно. Какие электрогенераторы из сети использовать, чтобы сум- марные затраты были минимальные, а потребность в электроэнергии бы- ла удовлетворена? Постройте математическую модель и найдите опти- мальное решение задачи.

1. Компания производит два вида полимерных материалов: полипропилен и полистирол. Для этого она нанимает неквалифицированных, квалифици- рованных и высококвалифицированных рабочих. Почасовая оплата в y.e. каждого рабочего и количество тонн материалов, производимых каждым рабочим в час, приводятся в табл. 19.

*Таблица 19*

Почасовая оплата и почасовые объемы производства

|  |  |  |  |
| --- | --- | --- | --- |
|  | Неквалифиц. | Квалифиц. | Высококвалифиц. |
| Оплата | 4 | 6 | 8 |
| Производительность (тонн в час) | | | |
| Полипропилена | 2 | 3 | 3 |
| Полистирола | 0 | 3 | 4 |

Рабочие могут производить одновременно оба продукта, например, ква- лифицированный рабочий может производить 3 тонны полипропилена и 3 тонны полистирола каждый час. Поступил заказ произвести за один час 21 тонну полипропилена и 15 тонн полистирола.

Требуется определить, сколько рабочих каждой квалификации необходи- мо нанять, чтобы выполнить заказ с наименьшими затратами на оплату труда. Постройте математическую модель и найдите оптимальное реше- ние.

Компания освоила производство поликарбоната и получила заказ про- извести 24 тонны за один час. При этом неквалифицированные рабочие могут произвести 1 тонну поликарбоната в час, квалифицированные ра- бочие — 3 тонны, а высококвалифицированным рабочим запрещено про- изводить поликарбонат. Как изменится модель при добавлении заказа на новый материал?

Компания получила разрешение на экспорт полистирола. Какую мини- мальную стоимость продажи за тонну следует установить компании, что- бы запуск производства дополнительных единиц этого материала был це- лесообразным? Постройте математическую модель. Найдите оптимальное решение задачи.

1. Производитель планирует запуск производства двух новых типов стекла: A и B. Для этого необходимо приобрести специальные печи. Стоимость

99

печи для производства стекла типа A составляет 500 тыс. у.е., стекла типа B — 600 тыс. у.е.

Для производства стекла необходимы песок, карбонат калия и карбонат кальция в заданных пропорциях (табл. 20).

*Таблица 20*

Содержание компонент в стекле

|  |  |  |  |
| --- | --- | --- | --- |
|  | Компоненты | | |
| Тип стекла | Песок | Карбонат калия | Карбонат кальция |
| A | 52 % | 13 % | 35 % |
| B | 73 % | 15 % | 12 % |

Согласно контракту поставщики могут доставить не более

1460 тонн песка, 500 тонн карбоната калия и 700 тонн карбоната кальция. Доход от продажи 1 тонны стекла типа A составляет 6 тыс. y.e., от про- дажи 1 тонны стекла типа B — 3 тыс. y.e. Производство устроено таким образом, что если оно запущено, то не менее одной тонны стекла любого типа должно быть произведено.

Постройте математическую модель, максимизирующую прибыль, исполь- зуя только следующие переменные:

*yA*(*yB*) =

для производства стекла типа A (B)

 0 − в противном случае,

 1*,* если покупается печь

*xA*(*xB*) — объем производства стекла типа A (B). Ответьте на следующие вопросы.

* 1. Какое из следующих ограничений входит в модель: 1.1) 0*.*52*xA* + 0*.*73*xb* ≥ 1460;

1.2) 0*.*52*xA* + 0*.*73*xb* ≤ 14*.*60;

1.3) 52*xA* + 73*xb* ≤ 1460;

1.4) 0*.*52*xA* + 0*.*73*xb* ≤ 1460?

* 1. Какое из следующих ограничений входит в модель: 2.1) *xA* − 1000*yA* ≤ 0;

2.2) 1000*xA* − *yA* ≤ 0;

2.3) *xB* − 1000*yB* ≥ 0?

* 1. Какое из следующих ограничений входит в модель:

100

3.1) *xA* − 1460*xA* ≥ 0;

3.2) 2000*yA* − *xA* ≥ 0;

3.3) 1460*xA* − *yA* ≤ 0;

3.4) *xA* − 1460*yA* ≤ 0?

1. Винодел, смешивая четыре сорта винограда, получает три сорта вина. Ко- личество виноградного сока, имеющегося в наличии (литров), пропорции смешивания и стоимость каждого сорта вина указаны в табл. 21.

*Таблица 21*

Данные о компонентах для производства вина

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сорт  вина | Сорт винограда | | | | Стоимость  вина  (за литр) |
| Алиготе | Изабелла | Рислинг | Шардоне |
| Каберне | \* | *≥* 75%  в любой пропорции | | *≥* 4% | 80 |
| Шардоне | \* | *≥* 10% | \* | *≤* 35% | 50 |
| Мускат | \* | *≥* 35% | | \* | 35 |
| наличие  (в литрах) | 130 | 200 | 150 | 350 |  |

∗ — в любом количестве.

Какие сорта винограда и в каком объеме необходимо смешивать, чтобы получить максимальный доход от продажи вина? Постройте математиче- скую модель и найдите оптимальное решение задачи.

1. В обязанности завхоза входит закупка продуктов питания, расфасован- ных в жестяные банки. По данным прошлого года, известно необходимое количество банок на каждый месяц (табл. 22).

*Таблица 22*

Ежемесячное необходимое количество банок

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сен | Окт | Ноя | Дек | Янв | Фев | Мар | Апр | Май |
| 1000 | 900 | 850 | 500 | 600 | 1000 | 1000 | 1000 | 500 |

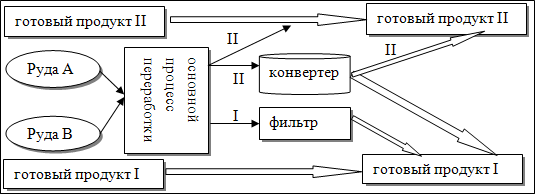
Завхоз может закупать продукты заранее или в начале каждого месяца по ценам, приведенным в табл. 23.

*Таблица 23*

Стоимость банки

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Сен | Окт | Ноя | Дек | Янв | Фев | Мар | Апр | Май |
| Цена  (y.e.) | 20 | 20 | 20 | 21 | 21 | 21 | 22 | 22 | 22 |

101



*Рис. 28.* Процесс переработки 1

Стоимость хранения составляет 0.2 y.e. за одну банку в месяц. В какие месяцы и в каком количестве завхоз должен делать закупки, чтобы обес- печить необходимым количеством продуктов с минимальными затраты на закупку и хранение. Постройте математическую модель и найдите оп- тимальное решение задачи.

1. Сантехник имеет большой запас 13-метровых медных труб. Ему нужны 10 труб длиной 4 метра, 10 труб длиной 5 метров и 23 трубы длиной 6 метров. Как он должен разрезать 13-метровые трубы? Постройте математическую модель минимизации числа использованных 13-метровых труб. Найдите оптимальное решение задачи.
2. Предприятие занимается переработкой руды. Процесс переработки схе- матически изображен на рис. 28. Перерабатываются два вида руды: А и В. Предприятию может быть поставлено до 100 тыс. тонн руды ви- да А по цене 3.25 y.e. за тонну и до 30 тыс. тонн руды вида В по цене

3.40 y.e. за тонну. Общая мощность основного процесса переработки рав- на 100 тыс. тонн руды при затратах на переработку 0.35 y.e. за тонну.

Основной процесс обработки позволяет получать

из каждой тонны руды вида А 0.15 тонн продукта I и 0.85 тонн про- дукта II,

•

из каждой тонны руды вида В 0.25 тонн продукта I и 0.75 тонн про- дукта II.

•

102

Продукт I более ценный, и агрегат, называемый конвертером, способен из каждой тонны продукта II получить 0.5 тонн продукта I и 0.5 тонн продук- та II, который нельзя повторно перерабатывать конвертером. Мощность конвертера — 50 тыс. тонн сырья при затратах на конвертерную обра- ботку 0.25 y.e. за тонну сырья. Затраты на фильтрацию продукта I после основного процесса обработки равны 0.10 y.e. тонн сырья.

Условия реализации продукции следующие. Вся продукция идет на про- дажу. Продукт II может быть реализован в неограниченном количестве по цене 3.80 y.e. за тонну. Продукт I продается по цене 5.50 y.e. за тонну, и его можно продать по этой цене до 45 тыс. тонн. Кроме того, можно про- дать до 4 тыс. тонн по цене 5.2 y.e. за тонну и неограниченное количество продукта по заниженной цене 5 y.e. за тонну.

Существует контракт, согласно которому требуется поставлять потреби- телям не менее 40 тыс. тонн продукта I. Оба продукта можно при необ- ходимости докупить: закупочная цена продукта I равна 5.75 y.e. за тонну, закупочная цена продукта II — 4 y.e. за тонну.

Постройте математическую модель, выполните следующие задания:

1. найдите план выпуска продукции с максимальной прибылью;
2. покажите, как изменится оптимальный план, если мощность филь- тра ограничена величиной 15 тыс. тонн;
3. покажите, как изменится оптимальный план производства, если

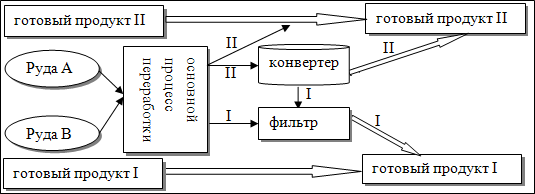
фильтр сломается и не будет задействован в производственном про- цессе, а продукт I в этом случае будет продаваться только по цене 5 y.e. за тонну?

Постройте математическую модель для задачи, в которой требуется най- ти единую (не зависящую от объемов продаж) минимальную цену на про- дукт I, при которой прибыль предприятия будет не меньше 50 тыс. y.e. По-прежнему ли это модель линейного программирования?

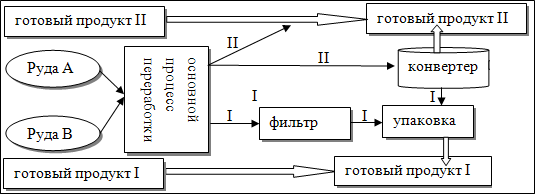
1. Выполните упражнение 53 со схемой работы предприятия, изображенной на рис. 29.
2. Выполните упражнение 53 со схемой работы предприятия, изображенной на рис. 30. Затраты на упаковку продукта I составляют 0.15 y.e. за тонну сырья.
3. На складе имеются прямоугольные листы материала размером 100 100 см2 в количестве 100 штук (рис. 31) и деловые отходы в форме мно- гоугольника в количестве 70 штук (рис. 32). В цех поступил заказ на изготовление деталей форм, указанных на рис. 33 в следующих количе- ствах: невыпуклых многоугольников — 550 штук; четвертей круга ради- усом 30 см — 800 штук; равнобедренных прямоугольных треугольников с

×

103

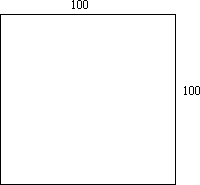


*Рис. 29.* Процесс переработки 2

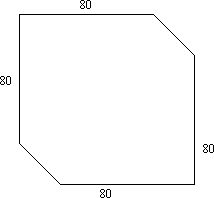


*Рис. 30.* Процесс переработки 3

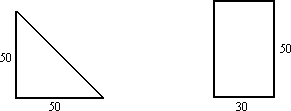
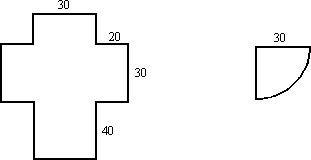
104



*Рис. 31.* Заготовки



*Рис. 32.* Деловые отходы



*Рис. 33.* Формы деталей в заказе

105

катетами 50 см — 330 штук; прямоугольников 50 30 см2 — 700 штук. При раскрое материала разрешаются произвольные повороты деталей. Най- дите, какое минимальное число листов материала нужно дополнительно закупить, чтобы выполнить заказ? Постройте математическую модель, решите и предъявите использованные карты раскроя.

×

1. Завод может выпускать насосы 10 типов: *V* 1, *V* 2, *SV* 1, *SV* 2, *SV* 3, *W* 1,

*W* 15, *W* 2, *SW* 2, *SW* 3. Для производства насосов каждого типа известны:

* + затраты (в тыс.y.e.) на наладку оборудования (табл. 24),

*Таблица 24*

Затраты на наладку оборудования

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* 1 | *SV* 2 | *SV* 3 | *W* 1 | *W* 15 | *W* 2 | *SW* | 2 | *SW* | 3 |
| 1 | 1 | 1.5 | 1.5 | 1.5 | 1.2 | 1.3 | 1.9 | 2 | 5 | | |

* + удельная стоимость (в y.e.) производства (табл. 25).

*Таблица 25*

Удельная стоимость производства

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* 1 | *SV* 2 | *SV* 3 | *W* 1 | *W* 15 | *W* 2 | *SW* 2 | *SW* 3 |
| 100 | 110 | 120 | 120 | 140 | 110 | 115 | 120 | 110 | 200 |

Насосы обладают свойством заменяемости. В каждом столбце *j* (табл. 26) в строке *i* указывается количество насосов типа *i*, способных заменить один насос типа *j*.

*Таблица 26*

Коэффициенты заменяемости насосов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V1 | | V2 | SV1 | SV2 | SV3 | W1 | W15 | W2 | SW2 | SW3 |
| V1 | 1 | ***2*** | 3 | - | - | 2 | 3 | - | - | - |
| V2 | 1 | 1 | 2 | 5 | - | - | - | - | - | - |
| SV1 | - | - | 1 | 4 | 10 | 1 | 2 | - | - | - |
| SV2 | - | 5 | 1 | 1 | 5 | - | - | - | 2 | 4 |
| SV3 | - | - | 1 | 1 | 1 | - | - | - | 2 | 2 |
| W1 | 1 | - | 1 | - | - | 1 | 1 | 2 | - | - |
| W15 | 1 | - | 1 | - | - | - | 1 | 2 | - | - |
| W2 | - | - | - | - | - | 1 | 1 | 1 | 2 | - |
| SW2 | - | 2 | - | 1 | - | - | - | 1 | 1 | 2 |
| SW3 | - | 1 | - | - | 1 | - | 1 | 1 | 1 | 1 |

символ «−» означает, что замена невозможна.

106

Например, выделенное жирным шрифтом число 2 в табл. 57 означает, что вместо одного насоса *V* 2 можно произвести 2 насоса *V* 1.

Решение о том, какие типы насосов будут заменены, принимается один раз до запуска производства. Каждый тип насоса можно заменить в со- ответствующей пропорции, указанной в таблице, только на один.

Требуется найти минимальный по затратам план выпуска насосов для выполнения заказа, указанного в табл. 27.

*Таблица 27*

Заказ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* | 1 | *SV* 2 | *SV* | 3 | *W* | 1 | *W* 15 | *W* 2 | *SW* | 2 | *SW* | 3 |
| 0 | 29 | 0 | 10 | | 0 | 0 | | 15 | | 20 | 0 | 0 | | |

Постройте математическую модель, решите задачу и ответьте на следу- ющие вопросы.

1. Как изменится план выпуска насосов, если суммарные затраты на наладку оборудования не должны превышать 3000 y.e.?
2. Изменится ли номенклатура выпускаемых насосов, если на складе имеется запас насосов типа *SW* 3 в количестве 50 шт.?
3. Известно, что для производства одного насоса требуется определен- ное количество килограмм резины указанное в табл. 28.

*Таблица 28*

Необходимое количество килограмм резины

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* | 1 | *SV* | 2 | *SV* | 3 | *W* | 1 | *W* 15 | *W* | 2 | *SW* | 2 | *SW* | 3 |
| 1 | 2 | 1 | 3 | | 1 | | 5 | | 1 | | 8 | 2 | | 7 | | |

Завод обладает запасом резины в размере 10 кг. Как изменится план выпуска, если завод может закупать резину по цене 20 y.e. за один килограмм?

1. Постройте математическую модель для случая, когда любой тип на- сосов можно заменять не более двумя (тремя) другими допустимыми типами.
2. Завод может выпускать насосы 10 типов: *V* 1, *V* 2, *SV* 1, *SV* 2, *SV* 3, *W* 1, *W* 15, *W* 2, *SW* 2, *SW* 3. Время на наладку оборудования (в тыс. мин.) при- водится в табл. 29.

*Таблица 29*

Время на наладку оборудования

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* 1 | *SV* 2 | *SV* 3 | *W* 1 | *W* 15 | *W* 2 | *SW* 2 | *SW* | 3 |
| 1 | 1 | 1.2 | 1.2 | 1.4 | 1.1 | 1.15 | 1.2 | 1.1 | 2 | |

107

Время, затрачиваемое на производство одного насоса в часах приводится в табл. 30.

*Таблица 30*

Время на производство

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* 1 | *SV* 2 | *SV* 3 | *W* 1 | *W* 15 | *W* 2 | *SW* 2 | *SW* 3 |
| 100 | 110 | 120 | 120 | 140 | 110 | 115 | 120 | 150 | 150 |

Насосы обладают свойством заменяемости. Условия заменяемости опи- саны в упражнении 57. Наладка оборудования для разных типов насосов начинается одновременно, начало производства разных типов насосов мо- жет осуществляться параллельно.

Требуется найти оптимальный по времени план выпуска насосов для вы- полнения заказа, указанного в табл. 31.

*Таблица 31*

Заказ насосов

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* | 1 | *SV* 2 | *SV* | 3 | *W* | 1 | *W* 15 | *W* 2 | *SW* | 2 | *SW* | 3 |
| 0 | 29 | 0 | 10 | | 0 | 0 | | 15 | | 20 | 0 | 0 | | |

Постройте математическую модель, решите задачу и ответьте на следу- ющие вопросы.

1. Как изменится план выпуска насосов, если суммарное время на на- ладку оборудования не должно превышать 3500 часов?
2. Изменится ли номенклатура выпускаемых насосов, если на складе имеется запас насосов типа *SW* 3 в количестве 50 шт.?
3. Известно, что наладка оборудования для производства каждого типа насосов требует финансовых затрат (тыс. y.e.), указанных в табл. 32.

*Таблица 32*

Финансовые затраты на наладку оборудования

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* 1 | *SV* 2 | *SV* 3 | *W* 1 | *W* 15 | *W* 2 | *SW* | 2 | *SW* | 3 |
| 1 | 1 | 1.5 | 1.5 | 1.5 | 1.2 | 1.3 | 1.9 | 2 | 5 | | |

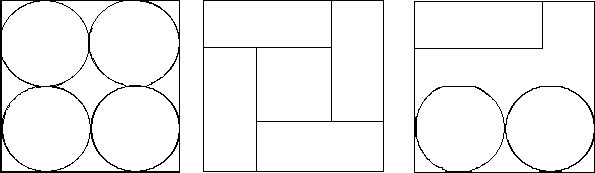
Удельная себестоимость (в y.e.) производства насосов каждого типа приводится в табл. 33.

*Таблица 33*

Удельная себестоимость производства насоса

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *V* 1 | *V* 2 | *SV* 1 | *SV* 2 | *SV* 3 | *W* 1 | *W* 15 | *W* 2 | *SW* 2 | *SW* 3 |
| 100 | 110 | 120 | 120 | 140 | 110 | 115 | 120 | 110 | 200 |

108



*Рис. 34.* Образцы раскроя

Как изменится план выпуска, если суммарные затраты на выполне- ние заказа не должны превышать 10000 y.e., а суммарное время на наладку оборудования не должно превышать 5000 часов?

1. Строительный магазин столкнулся со следующей проблемой. Клиентам нужен оргалит размером 120 90 см2, 120 150 см2 и 120 180 см2 в количестве не менее 20, 50 и 40 листов по цене 700, 900 и 1000 y.e. за один лист соответствующего размера. Магазин может вырезать листы нужного размера из стандартных листов большего размера 120 240 см2, которых имеется неограниченное количество, оптовая стоимость закупки таких листов составляет 600 y.e. за один лист. Стоимость выполнения одного разреза составляет 150 y.e.

×

× × ×

Каким образом следует разрезать большие листы, чтобы получить мак- симальную прибыль? Постройте математическую модель, решите задачу и предъявите схемы разрезов.

1. У раскройщика имеется 30 листов фанеры размера 300 300 см2, из кото- рых необходимо выкроить 20 кругов диаметром 150 см и 15 прямоуголь- ников размером 120 180 см2. Образцы раскроя приводятся на рис. 34. Стоимость выкраивания одного круга составляет 200 y.e., одного прямо- угольника — 150 y.e. В стоимость включено выполнение всех разрезов, и она не зависит от выбранного образца. Круги и прямоугольники мож- но купить готовые по цене 450 и 300 y.e. соответственно, если не хватит больших листов. Раскройщик стремится минимизировать расходы на вы- краивание. При этом он должен получить нужное количество кругов и прямоугольников и использовать больших листов не больше, чем у него есть, а суммарные отходы должны быть не более 30 %. Постройте мате- матическую модель и найдите решение.

×

×

109

1. Менеджер компании «K&Co» должен выбрать, в какие из пяти проек- тов *A*, *B*, *C*, *D* или *E* сделать инвестиции на ближайшие 3 года так, чтобы получить максимальную прибыль. Проекты могут принести доход (млн y.e.), указанный в табл. 34.

*Таблица 34*

Доход от проектов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *A* | *B* | *C* | *D* | *E* |
| 3 | 2 | 1 | 4 | 2 |

Менеджер обратился к аналитикам и получил следующие рекомендации.

* 1. По крайней мере в один из проектов *C*, *D* или *E* обязательно нужно сделать инвестиции, так как они идеально соответствуют специали- зации сотрудников компании.
  2. Если для инвестиций выбран проект *A*, то и проект *D* тоже должен быть выбран.
  3. Если выбраны проекты *A* и *B*, то в проект *C* не нужно инвестировать из-за высоких рисков.
  4. Если выбраны проекты *B* и *C*, то нужно инвестировать и в *E*, по- скольку в этом случае можно получить хорошие скидки от постав- щиков.

Постройте математическую модель и найдите оптимальное решение за- дачи.

1. В 2015 году Алиса планирует отправиться в кругосветное путешествие. В начале 2011 года она готова начать делать сбережения для путешествия так, чтобы к началу 2015 года у нее было 21 тыс. евро.

Она может выбирать из трех типов акций *A*, *B* или *C*. Каждая акция стоит 100 y.e. Покупка акций осуществляется в начале каждого года. Че- рез год инвестиций в акции типа *A* выплачивается 104 y.e., через два года инвестиций в акции типа *B* — 110 y.e., через четыре года инвестиций в акции типа *C* — 125 y.e. В начале каждого года — 2011, 2012, 2013 и 2014 Алиса планирует инвестировать всего не более 5 тыс. y.e.

Необходимо найти минимальный суммарный объем инвестиций за 2011– 2014 годы. Постройте математическую модель и найдите оптимальный план.

Как изменится модель, если появляется еще один тип инвестиций: акции типа *D*? Стоимость каждой такой акции составляет 100 y.e., а через три года ее стоимость составит 115 y.e.

110

1. Завод производит некоторую продукцию. Клиенты сразу забирают её с места производства, так как у завода нет складских помещений. Если же объемы производства превысили спрос, то продукция отправляется на склад магазина. Если же произведенной продукции недостаточно для удовлетворения спроса, то недостающее количество нужно привести со склада магазина. Продукция отправляется на склад или забирается со склада в конце дня, а поступает на склад или к клиентам в начале сле- дующего дня. Удельные транспортные затраты на перевозку продукции с или на склад составляют 0.2 y.e.

Известно, что в первый день на складе лежит 100 единиц продукции, к концу 4-го дня там должно оставаться 100 единиц продукции. Ежеднев- ный спрос и максимальный объем производства в единицах приводятся в табл. 35.

*Таблица 35*

Ежедневный спрос и максимальный объем производства

|  |  |  |
| --- | --- | --- |
| День | Спрос | Производство |
| первый | 60 | 70 |
| второй | 80 | 50 |
| третий | 70 | 90 |
| четвертый | 50 | 70 |

Удельные затраты на производство составляют 5 y.e. в день, удельная стоимость хранения — 0.1 y.e. в день.

Необходимо составить план производства, перевозки и управления запа- сами продукции, чтобы удовлетворить спрос на 4 дня с минимальными суммарными затратами. Постройте математическую модель и найдите оп- тимальный план инвестиций.

1. Фирма специализируется на производстве высококачественных горных велосипедов одной модели. Спрос велосипедов в каждом месяце ограни- чен. Каждый раз, чтобы запустить производство, необходимо оплатить наладку оборудования. Производить можно не более одной партии в ме- сяц. Предполагается, что производственные мощности фирмы неограни- чены.

Затраты на наладку производства составляют 5000 y.e., стоимость произ- водства одной единицы составляет 100 у.е. Таким образом, производство партии из 1 велосипеда требует затрат в 5100 у.е., для производства пар- тии из 10 велосипедов затраты составят 5000 + (10 · 100)= 6000 y.e.

В табл. 36 приводится прогноз ежемесячного спроса *dt* на велосипеды на следующий год.

111

*Таблица 36*

Ежемесячный спрос

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Янв | Фев | Мар | Апр | Май | Июн |
| 400 | 400 | 800 | 800 | 1200 | 1200 |
| Июл | Авг | Сен | Окт | Ноя | Дек |
| 1200 | 1200 | 800 | 800 | 400 | 400 |

Известно, что на складе осталось 200 велосипедов с прошлого года. Удель- ная стоимость хранения в месяц составляет 5 y.e., вместимость склада неограничена. Задача менеджера — составить такой план производства и хранения велосипедов, чтобы удовлетворить спрос с минимальными сум- марными затратами на год.

65. Проект состоит из 8 работ. Длительности выполнения каждой работы (дней) приводятся в табл. 37.

*Таблица 37*

Длительность выполнения работы

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер работы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Длительность | 3 | 2 | 4 | 3 | 2 | 1 | 5 | 4 |

Работы можно выполнять на несколько дней быстрее, но не быстрее, чем на максимальное количество дней, приведенное в табл. 38.

*Таблица 38*

Максимальное сокращение дней

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер работы | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Кол-во | дней | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 |

Стоимость ускорения выполнения работы на один день (в y.e.) приводится в табл. 39.

*Таблица 39*

Стоимость ускорения выполнения работы

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| номер работы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| стоимость | 10 | 10 | 12 | 11 | 11 | 12 | 13 | 13 |

У некоторых работ имеются непосредственные работы-предшественники (табл. 40).

112

*Таблица 40*

Отношения предшествования

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер раб. | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Номер | раб.-предшест. | — | — | 1,2 | 1,2 | 3,4 | 3,4 | 3 | 5,6 |

Постройте сеть (работы – дуги) и выберите правильные варианты ответов на следующие вопросы.

Наиболее раннее время завершения всего проекта равно

1) 11;

2) 12;

3) 13;

4) 14.

Наиболее позднее время завершения работы 4 равно 1) 6;

2) 7;

3) 8;

4) 9.

Какой путь является критическим в данном проекте: 1) 1-3-7;

2) 1-3-6-8;

3) 1-4-5-8;

4) 1-3-5-8?

Постройте математическую модель и найдите минимальные суммарные затраты при условии, что проект должен быть завершен за 10 дней.

66. На прошлой неделе Ксения, Алексей, Степа, Наталья и я каждый вечер с понедельника по пятницу ужинали вместе. Каждый из нас в начале каждого дня по очереди выбирал подходящее место. В пятницу Наталья пропустила ужин с друзьями, так как навещала родственников, но, тем не менее, выбрала подходящее место для друзей. Мы посетили следующие места: французский ресторан, суши-бар, пиццерию, греческий ресторан и пивоварню. Ксения повела нас в среду. Пятничный ужин был в пивоварне. Степа не любит суши, и он первый выбирал место для ужина. Я выбрала французский ресторан, а на следующий день один из друзей повел всех в пиццерию. Ответьте на вопрос, кто, когда и какое место для ужина выбрал? (Используйте задачу о выполнимости.)

113

1. Пять парикмахеров решили подстричь друг друга. В табл. 41 приводится время в минутах, затрачиваемое на стрижку.

*Таблица 41*

Продолжительность стрижки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Елена | Максим | Наталья | Ирина | Ольга |
| Елена | – | 42 | 37 | 36 | 44 |
| Максим | 31 | – | 43 | 48 | 45 |
| Наталья | 43 | 34 | – | 27 | 44 |
| Ирина | 26 | 28 | 44 | – | 40 |
| Ольга | 47 | 43 | 42 | 31 | – |

Составьте математическую модель и найдите назначения между парик- махерами так, чтобы подстричь всех парикмахеров за наименьшее общее время.

1. На склад экспресс-почты прибыло восемь посылок, которые требуется до- ставить восьми адресатам. За хранение каждой посылки в течение одного часа взимается плата в размере *ci* y.e., время доставки посылки *i* до ад- ресата составляет *ti* часов, *i* = 1*, . . . ,* 8, данные приводятся в табл. 42.

*Таблица 42*

Время доставки посылок

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| *ci* | 8 | 8 | 8 | 4 | 5 | 8 | 7 | 2 |
| *ti* | 11 | 14 | 10 | 16 | 19 | 18 | 16 | 19 |

В службе доставки работает только один курьер. Постройте математи- ческую модель и найдите, в какой последовательности нужно доставить посылки, чтобы суммарная стоимость хранения была минимальной.

1. Завод специализируется на производстве продукции из стекла. Установ- лено три производственных линии для производства шести типов продук- тов, которые нужно произвести за 15 дней. На завод поступает заказ на разные типы продуктов. Необходимо составить план выполнения заказа с минимальными затратами на производство и хранение продукции.

Единовременно каждая линия может производить только один тип про- дукта. Каждый тип продукта в каждый момент времени производится только на одной линии. Операции на линии выполняются целое число дней. Линия работает непрерывно без остановок. Переход с производства одного продукта на другой для каждой линии осуществляется в начале дня. При этом некоторое время, пока оборудование перенастраивается,

114

производственные мощности падают. В это время производится продукт низкого качества, который потом уходит в отходы. Объемы некачествен- ного продукта зависят от типов продуктов, между которыми был переход, и от производственной линии. Скорость производства каждой линии по- стоянна и не зависит ни от типа продукта, ни от величины заказа на него.

Суммарные затраты складываются из затрат на хранение продукции и производственных затрат. Известны ежедневные производственные за- траты на каждый тип продукта на любой линии. В начале периода пла- нирования готовой продукции нет.

В день можно произвести не более 342 единиц каждого типа продукта на каждой линии. Ежедневная удельная стоимость хранения (в y.e.) каждого типа продукта приводится в табл. 43.

*Таблица 43*

Ежедневная удельная стоимость хранения

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Продукт | | 1 | 2 | 3 | 4 | 5 | 6 |
| Стоимость | хранения | 5 | 5 | 1 | 3 | 2 | 1 |

Ежедневная удельная стоимость производства (в y.e.) каждого продукта на любой линии приводится в табл. 44.

*Таблица 44*

Ежедневная удельная стоимость производства

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Продукт | 1 | 2 | 3 | 4 | 5 | 6 |
| Стоимость производства | 800 | 800 | 600 | 400 | 400 | 800 |

*Таблица 45*

Ежедневный спрос на каждый тип продукта

115

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| День | 1 | Величина  на каждый  2 3 | | | спроса  продукт  4 5 | | 6 |
| 1 | 0 | 85 | 0 | 102 | | 0 | 89 |
| 2 | 76 | 0 | 77 | 0 | | 111 | 0 |
| 3 | 116 | 95 | 124 | 92 | | 87 | 0 |
| 4 | 102 | 0 | 106 | 0 | | 94 | 83 |
| 5 | 109 | 101 | 79 | 118 | | 91 | 86 |
| 6 | 100 | 120 | 123 | 108 | | 90 | 118 |
| 7 | 93 | 99 | 107 | 95 | | 111 | 108 |
| 8 | 98 | 120 | 85 | 78 | | 106 | 83 |
| 9 | 93 | 114 | 98 | 108 | | 117 | 79 |
| 10 | 118 | 94 | 105 | 84 | | 105 | 80 |
| 11 | 83 | 78 | 119 | 111 | | 91 | 99 |
| 12 | 122 | 91 | 98 | 121 | | 97 | 92 |
| 13 | 77 | 86 | 125 | 122 | | 107 | 98 |
| 14 | 123 | 104 | 114 | 102 | | 83 | 78 |
| 15 | 125 | 97 | 109 | 80 | | 103 | 95 |

Количество единиц продукции *j*-го типа низкого качества при переклю-

чении с *i*-го продукта на *k*-й линии рассчитывается по формуле *cij* =

*k*

*sti* + 10(*k* 1) + 10 *j i* , где *st* = (40*,* 20*,* 20*,* 20*,* 40*,* 50) — начальное количе- ство низкокачественного продукта. После переключения линии на новый продукт (пока производится низкокачественый) производственные затра- ты считаются для нового продукта. В начале горизонта планирования все линии настроены на производство качественного продукта. Данные о спросе на каждый тип продукта на каждый день приводятся в табл. 45. Постройте математическую модель и найдите оптимальное решение за- дачи.

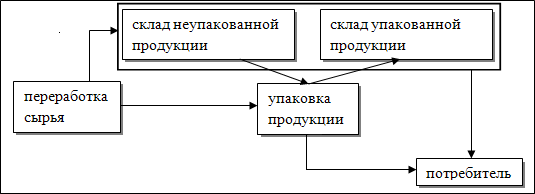
− | − |

70. Предприятие установило систему производственного планирования в сво- их отделениях. Задача системы — осуществлять планирование производ- ства и хранения продукта. Производственный процесс состоит из несколь- ких шагов: переработка сырья, упаковка продукта, хранение и отправка упакованного продукта потребителям. Планирование осуществляется на 3 месяца.

Схема производственного процесса представлена на рис. 35. Процесс про- изводства начинается с переработки сырья, количество которого неогра- ничено. В результате получается 4 типа продукта: *A*, *B*, *C*, *D*. Ежемесячно каждого продукта можно произвести не более *Ci* штук, *i* = *A, B, C, D*. Известна потребность *Dt* штук *i*-го типа продукта в месяц *t*. После перера- ботки сырья продукт отправляется либо в цех по упаковке, либо на склад. Вместимость склада неограничена. Цех по упаковке может обрабатывать не более 28000 штук продукции в месяц. Упакованный продукт может отправляться либо на склад, либо к потребителям. На складе имеются

*i*

116



*Рис. 35.* Схема производственного процесса

помещения для хранения упакованной и неупакованной продукции. По- скольку для хранения упакованной продукции нужны специальные стел- лажи, то затраты на хранение упакованного продукта на 4 y.e. за штуку в месяц больше, чем затраты на хранение неупакованного продукта. Дан- ные приводятся в табл. 46.

*Таблица 46*

Затраты на хранение неупакованного продукта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *i* | *Dt* (шт.)  *i* | | | *Ci*  (шт. в мес.) | Затраты  на хранение  неупакованного продукта  (y.e. за шт. в мес.) |
| *t* = 1 | *t* = 2 | *t* = 3 |
| A  B C  D | 5000  900  6000  10000 | 6000  1000  9000  11000 | 3000  4000  4000  14000 | 8000  5000  8000  15000 | 35  39  45  85 |

1. Какое количество продукции необходимо производить ежемесячно, что- бы удовлетворить запросы потребителей с наименьшими суммарными за- тратами?
2. Стоит ли увеличивать мощность цеха по упаковке? Какие данные Вам необходимы, чтобы ответить на этот вопрос?
3. При смене продукта необходимо очищать упаковочную линию, а для этого используются дорогостоящие препараты. В начале каждого меся- ца, кроме первого, линию нужно очищать, если тип продукта меняется. Затраты на очистку упаковочной линии перед упаковкой каждого типа продукта приводятся в табл. 47.

117

*Таблица 47*

Затраты на очистку упаковочной линии

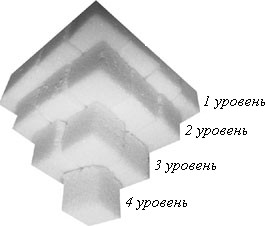
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Продукт | *A* | *B* | *C* | *D* |
| Стоимость очистки (тыс. y.e.) | 500 | 900 | 800 | 900 |

Найдите новый план производства и хранения продукции, доставляющий минимальные суммарные затраты. Постройте математическую модель и найдите оптимальное решение.

71. Промышленное предприятие получило разрешение на разработку место- рождения руды на участке размером 200 200 футов2. Вскрытие участка производится наклонными выработками, предельный уклон наклона 450.

×

С учетом максимального угла наклона и с целью предотвращения засы- пания нижних слоев и заполнения всего поперечного сечения выработки выемку руды делают блоками высотой 25 футов, шириной и длиной по 50 футов, размещая их, как показано на рис. 36. Чтобы добраться до внутреннего блока, не лежащего на поверхности, нужно отработать «пи- рамиду» из четырех блоков, лежащих на один слой выше этого блока. Проведенные исследования местности позволили оценить возможное со-



*Рис. 36.* Выработка внутреннего блока

держание руды (в процентах) в разных местах участка на разной глубине для каждого блока (табл. 48—51).

*Таблица 48*

Содержание руды на 1 уровне

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 уровень  (поверхность) | 1.5 | 1.5 | 1.5 | 0.75 |
| 1.5 | 2.0 | 1.5 | 0.75 |
| 1.0 | 1.0 | 0.75 | 0.5 |
| 0.75 | 0.75 | 0.5 | 0.25 |

118

Содержание руды на 2 уровне

|  |  |  |  |
| --- | --- | --- | --- |
| 2 уровень  (глубина 25 футов) | 4.0 | 4.0 | 2.0 |
| 3.0 | 3.0 | 1.0 |
| 2.0 | 2.0 | 0.5 |

Содержание руды на 3 уровне

|  |  |  |
| --- | --- | --- |
| 3 уровень  (глубина 50 футов) | 12.0 | 6.0 |
| 5.0 | 4.0 |

Содержание руды на 4 уровне

*Таблица 49*

*Таблица 50*

*Таблица 51*

6.0

4 уровень

(глубина 75 футов)

Если бы блок состоял на 100 % из руды, то он приносил бы доход в 200 тыс. y.e.

Стоимость выработки одного блока зависит от глубины (табл. 52).

*Таблица 52*

Стоимость выработки одного блока на разной глубине

|  |  |  |
| --- | --- | --- |
| Уровень | Стоимость (тыс. | y.e.) |
| 1 | 3 | |
| 2 | 6 | |
| 3 | 8 | |
| 4 | 10 | |

Какие блоки следует вырабатывать, чтобы получить максимальную при- быль? Постройте математическую модель и найдите оптимальное реше- ние. В модели должно быть 56 ограничений и 30 переменных.

Можно ли использовать методы решения задач линейного программиро- вания для поиска оптимального решения?

72. Предприятие занимается переработкой неочищенных масел. Перед тем, как получить готовый продукт, сырье проходит две стадии обработки. Сначала масла очищаются, а затем смешиваются. В наличии имеется пять типов масел: два из которых растительного происхождения *V* 1, *V* 2, три — животного *O*1, *O*2, *O*3. Горизонт планирования производства — с января по июнь. Сырье можно покупать как непосредственно перед использова- нием в любой месяц горизонта планирования, так и заранее. В табл. 53 приводится стоимость покупки тонны масла в соответствующий месяц.

119

*Таблица 53*

Стоимость тонны масла

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *V* 1 | *V* 2 | *O*1 | *O*2 | *O*3 |
| Январь | 110 | 120 | 130 | 110 | 115 |
| Февраль | 130 | 130 | 110 | 90 | 115 |
| Март | 110 | 140 | 130 | 100 | 95 |
| Апрель | 120 | 110 | 120 | 120 | 125 |
| Май | 100 | 120 | 150 | 110 | 105 |
| Июнь | 90 | 100 | 140 | 80 | 135 |

Конечный продукт продается по цене 150 y.e. за тонну.

Очистка растительных и животных масел выполняется на разных аппара- тах. Согласно технологическим требованиям аппараты ежемесячно могут очищать не более 200 тонн растительного масла, и не более 250 тонн мас- ла животного происхождения. В процессе очистки сырье не теряет в весе. Затратами на очистку можно пренебречь.

На складе можно хранить всего до 1000 тонн неочищенного растительного и столько же животного происхождения масел. Стоимость хранения лю- бого масла 5 y.e. за тонну в месяц. Масло, прошедшее очистку, и готовый продукт не могут храниться на складе. В начале января на складе лежит 500 тонн масла каждого происхождения и столько же должно остаться к концу горизонта планирования, т. е. в конце июня.

Качество готового продукта оценивается его плотностью. Этот показа- тель зависит прямо пропорционально от массы (и только от нее) и дол- жен принимать значение от 3 до 6. Плотность масел приводится в табл. 54. Постройте математическую модель и определите, какие масла и в каком количестве должен покупать производитель, чтобы получить максималь- ную прибыль.

*Таблица 54*

Плотность масел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *V* 1  8.8 | *V* 2  6.1 | *O*1  2.0 | *O*2  4.2 | *O*3  5.0 |

1. Модифицируйте модель для задачи 72, учитывая следующие требования.
   1. При изготовлении продукта в смеси всегда должно присутствовать не более трех различных типов масел.
   2. Нельзя использовать меньше 20 тонн масла любого типа в месяц.
   3. Если при изготовлении продукта в смеси присутствует масло *V* 1 или

*V* 2, то обязательно должно быть и масло *O*3.

120

74. Крупная компания собирается принять решение о переносе некоторых филиалов из Лондона в другие города. Это позволит сотрудникам ком- пании приобрести недорогое жилье, обеспечить рабочими местами жи- телей других городов. Новые филиалы могут размещаться в Эдинбурге, в Брайтоне либо остаться в Лондоне. Необходимо разместить пять фили- алов. В каждом городе может быть расположено не более трех филиалов. Выгода (тыс. y.e. в год) от перемещения филиалов в другие города при- водится в табл. 55.

*Таблица 55*

Выгода от перемещения филиалов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| Эдинбург | 10 | 15 | 10 | 20 | 5 |
| Брайтон | 10 | 20 | 15 | 15 | 15 |

При размещении в разных городах возникают затраты на сообщение меж- ду филиалами, которые зависят от местоположения филиалов и объемов продукции, пересылаемых из одного филиала в другой. В табл. 56 приво- дятся объемы (тыс. единиц) пересылаемые, между филиалами.

*Таблица 56*

Объемы пересылки между филиалами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A B C D | A  0  0  0  0 | B  0  0  0  0 | C  1  1.4  0  0 | D  1.5  1.2  0  0 | E  0  0  2  0.7 |

В табл. 57 приводятся стоимости пересылки единицы продукции в год (y.e.).

*Таблица 57*

Удельная стоимость пересылки

|  |  |  |  |
| --- | --- | --- | --- |
|  | Эдинбург | Брайтон | Лондон |
| Эдинбург | 5 | 14 | 13 |
| Брайтон | 0 | 5 | 9 |
| Лондон | 0 | 0 | 10 |

Где нужно разместить филиалы, чтобы минимизировать ежегодные об- щие расходы? Постройте математическую модель целочисленного линей- ного программирования и найдите оптимальное решение.

121

1. Крупная кинокомпания планирует съемку фильма. Известно, что в съем- ке фильма будут участвовать *n* актеров. Фильм состоит из *m* эпизодов. Каждый день снимается ровно один эпизод. Для каждого эпизода из- вестны актеры, которые в нем задействованы. Актер может сниматься в нескольких эпизодах. Он приезжает на съемочную площадку в тот день, когда снимается первый эпизод с его участием, и покидает площадку в конце последнего дня съемок всех эпизодов с его участием. За каждый день присутствия на съемочной площадке (независимо от того, занят ли актер в съемке в этот день или нет) кинокомпания выплачивает ему пер- сональный гонорар.

Найдите, в каком порядке должны сниматься эпизоды, чтобы сумма вы- плаченных гонораров была минимальной.

Постройте математическую модель и найдите оптимальное решение для съемок фильма «Новосибирск, я тебя люблю!».

Эпизоды фильма:

* 1. Знакомство с городом. Участвуют: Брюс, Джеки, Брэд, Том и Пенело- па.
  2. У фонтана. Участвуют: Брэд, Джулия и Сальма.
  3. В парке. Участвуют: Брэд, Том и Джулия.
  4. Знакомство с университетом. Участвуют: Брюс, Джеки и Пенелопа.
  5. Прогулка. Участвуют: Брэд и Сальма.
  6. В театре. Участвуют: Брюс, Брэд, Том, Джулия и Пенелопа.

Персональные гонорары актеров (в тыс. y.e. за один день) приводятся в табл. 58.

*Таблица 58*

Персональные гонорары актеров

Брюс Джеки Брэд Том Пенелопа Джулия Сальма 100 70 65 75 110 130 120

122

# 6. Решение оптимизационных задач в GAMS

В этом разделе на одном из примеров, рассмотренном в начале пособия, будет разобран процесс построения модели в системе GAMS.

GAMS — это аббревиатура английских слов General Algebraic Modeling System, что можно перевести как «система для алгебраического моделирова- ния». Система GAMS представляет собой оболочку для описания моделей. Она не решает задачу, а направляет к одной из решающих программ (solver или ре- шатель) CPLEX, BARON и др.

*Пример*

Фирма производит шесть типов стульев: *офисный*, *помощник*, *капитан*, *мар- киза*, *испанский* и *венский*. Для производства стульев необходимы следующие детали: длинные и короткие болты, тяжелые и легкие сиденья, длинные и ко- роткие ножки, перекладины и др. В табл. 59 приводятся данные о стоимости стульев, потребности в деталях для каждого типа стула и их наличие на складе.

*Таблица 59*

Количество деталей на складе

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Тип стула | *Офис* | *Пом* | *Кап* | *Марк* | *Исп* | *Вен* | |
| Цена | 36 | 40 | 45 | 38 | 35 | 25 |  |
| Требуемое количество деталей | | | | | | | Всего  на складе |
| *длин. болты*  *корот. болты тяж. сиденья лег. сиденья длин. ножки корот. ножки перекладины гайки*  *роллеры*  *каркас крепления* | 8  4  4  1  0  6  0  1  0  0  0 | 0  12  4  0  1  0  4  0  1  0  0 | 12  0  4  0  1  4  0  0  0  1  0 | 0  12  4  0  1  0  5  0  0  1  0 | 8  4  4  1  0  5  0  0  0  0  1 | 4  8  4  1  0  0  6  0  0  0  1 | 1280  1900  1090  190  170  1000  1000  110  72  93  85 |

Необходимо определить, какие стулья и в каком количестве нужно произво- дить, чтобы получить максимальный доход, используя имеющееся количество деталей на складе. Введем обозначения:

*I* — множество типов стульев, *I* = *кап.*, *пом.*, *мар.*, *исп.*, *вен.*, *офис.* ;

{ }

*J* — множество деталей, *J* = *дл.бол.*, *кор.бол.*, *тяж.сид.*, *лег.сид.*, *длин.нож.*, *кор.нож.*, *перек.*,*гайк.*, *ролл.*, *кар.*, *креп.* ;

}

{

*pi* — цена *i*-го типа стула, *i I*;

∈

*dji* — количество деталей *j*-го типа, необходимое для производства одного сту- ла *i*-го типа, *i I*, *j J* ;

∈ ∈

*qj* — запас деталей *j*-го типа на складе, *j J* . Введем переменные:

∈

123

*xi* — количество стульев *i*-го типа, которое производит фирма;

*z* — суммарный доход от продажи стульев. С точки зрения моделей целочислен- ного линейного программирования необходимости в такой переменной нет. Она необходима для работы с моделью в GAMS. Построим математическую модель:

max *z*

*z* = *pixi function*

∑

*i*∈*I*

∑

*djixi* ≤ *qj, j* ∈ *J constraints1*

*i*∈*I*

*xi* ≥ 0*,* целые*, i* ∈ *I constraints2*

Целевая функция — максимизировать суммарный доход (*function*). Группа ограничений *constraints1* гарантирует, что имеющихся деталей на складе доста- точно для производства стульев. Группа ограничений *constraints2* определяет допустимые значения переменных.

Ниже приводится текст программы для решения этой задачи в GAMS.

1. $ontext
2. Решается задача о работе мебельного комбината.
3. Математическая модель записана
4. в терминах целочисленного линейного программирования.
5. $offtext
6. Sets
7. I множество стульев /1\*6/
8. J множество деталей /1\*11/ 9.;
9. Parameters
10. p(i) стоимость i-го типа стула i из I 12. /1 36

13. 2 40

14. 3 45

15. 4 38

16. 5 35

17. 6 25 /

18. q(j) кол-во деталей j-го типа на складе j из J 19. /1 1280

20. 2 1900

21. 3 1090

22. 4 190

23. 5 170

124

24. 6 1000

25. 7 1000

26. 8 110

27. 9 72

28. 10 93

29. 11 85 /;

1. Table
2. d(j,i) кол-во деталей j-го типа для пр-ва стула i-го типа 32. 1 2 3 4 5 6

33. 1 8 0 12 0 8 4

34. 2 4 12 0 12 4 8

35. 3 4 4 4 4 4 4

36. 4 1 0 0 0 1 1

37. 5 0 1 1 1 0 0

38. 6 6 0 4 0 5 0

39. 7 0 4 0 5 0 6

40. 8 1 0 0 0 0 0

41. 9 0 1 0 0 0 0

42. 10 0 0 1 1 0 0

43. 11 0 0 0 0 1 1 ;

1. Integer Variables
2. x(i)сколько стульев $i$-го типа производит фирма;
3. x.up(i)=INF;
4. Free Variable
5. z суммарный доход от продажи стульев;
6. Equations
7. function суммарный доход от продажи
8. constraints1(j) ограничения на кол-во деталей на складе;
9. function.. z =e= sum(i,x(i)\*p(i));
10. constraints1(j).. sum(i,d(j,i)\*x(i)) =l= q(j);
11. Model Chairs /function,constraints1/;
12. \*Model Chairs /all/;
13. Chairs.optcr=0.0;
14. Solve Chairs using mip max z;
15. Display x.l, z.l, constraints1.l;

Программа состоит из трех частей. Первая часть — описание исходных дан- ных задачи, вторая часть — описание переменных, декларирование ограниче- ний и непосредственное описание модели (целевой функции и ограничений), третья часть — обращение к решателю и выдача результатов.

Строки 1—5 соответствуют комментариям. Для того, чтобы закомментиро- вать одну строку, нужно поставить знак « » в начале строки. Для того, чтобы закомментировать несколько строк, нужно окружить их командами $*ontext* и

∗

$*of f text*.

125

Строки 6—42 соответствуют описанию исходных данных задачи. Для этого используются специальные блоки: *Sets* — для задания множеств, *P arameters*

— для векторных величин, *T ables* — для матриц, *Scalars* — для скалярных величин и констант. Эти блоки могут следовать в произвольном порядке друг за другом, но должны при этом использовать определенные ранее величины.

Для сокращения кода программы типы стульев *капитан*, *помощник*, *мар- киза*, *испанский*, *венский* и *офисный* пронумерованы подряд от 1 до 6, а детали пронумеровали подряд от 1 до 11. В строке 7 приводится задание множества стульев *I*, в строке 8 — задание множества деталей *J* . Между именем множества и перечислением элементов, входящих в это множество, находится поясняющий текст.

В именах множеств, таблиц, векторов, переменных, ограничений и пр. нель- зя использовать кириллицу. В конце каждого блока ставится знак «;», строка 9. Строки 10—29 соответствуют описанию векторов *pi* и *qj*. Между двумя на-

клонными чертами «*/ . . . /*» заключены пары: координата вектора и значение по этой координате. В конце блока ставится «;».

Строки 30—43 соответствуют описанию матрицы (*dji*). Обратите внимание, что каждый столбец в таблице выровнен по левому (допускается выравнивание по правому) краю.

В строках 44—48 декларируются переменные. В задаче имеется шесть це- лочисленных неотрицательных переменных *x*(*i*). На языке GAMS они объяв- ляются Integer Variables. По умолчанию диапазон значений переменных типа Integer от 0 до 100. В данном примере это ограничение может оказаться суще- ственным, поэтому необходимо расширить диапазон Integer. Для этого в стро- ке 46 изменяется максимальное значение переменных *x*(*i*) с помощью команды *x.up*(*i*) = *IN F* , где INF — это некоторое большое число в GAMS, аналог бес- конечности. Переменная *z*, которая соответствует значению целевой функции, должна быть свободной. На языке GAMS она объявляется Free Variable («Free» можно опустить).

Строки 49—53 соответствуют блоку Equations. В нем объявляются имена всех ограничений, чтобы в дальнейшем на них ссылаться, и выписываются ограничения. Символы две точки подряд «..» всегда разделяют название огра- ничения и начало алгебраического выражения. Символы = *e* = означают ра- венство (от англ. equal — равно); = *l* = означают меньше либо равно (от англ. less — меньше); = *g* = означают больше, либо равно (от англ. greater — больше).

После того, как все ограничения описаны, необходимо указать, какие из них нужно учитывать при решении задачи. Возможно, что какие-то ограничения оказались несущественными, или Вы хотите решить задачу несколько раз с разными наборами ограничений.

В строке 54 с помощью оператора *M odel* присваивается название модели и перечисляются именно те ограничения, которые следует учесть при решении задачи. В данном примере модель называется *Chairs*. Между двумя наклон- ными чертами */ . . . /* перечислены те ограничения, которые вошли в модель.

126

Поскольку все описанные ограничения формируют модель, то можно записать это в более компактной форме с помощью ключевого слова *all* (от англ. all — все), строка 55.

Строки 56, 57 соответствуют подготовке и обращению к решателю. Для того, чтобы вызвать решатель, нужно в операторе *Solve* указать название ре- шаемой модели, её тип и критерий оптимизации. Модель *Chairs* относится к целочисленному линейному программированию. Критерий оптимизации — максимизация дохода *z*.

Для задач минимизации используется ключевое слово min. Кроме задач класса *mip*, можно решать и другие. Это зависит от возможностей установ- ленного решателя. В меню *F ile* −−· *Options* −−· *Solvers* можно подключить

соответствующий решатель.

Перед тем, как вызывать решатель, можно сделать ряд настроек, улучшаю- щих качество получаемого решения. Так, в строке 56 устанавливается относи- тельный разрыв (*relativegap*), равный 0 %, который по умолчанию составляет 10 %.

В результате выполнения оператора *Solve*:

* подготовлены данные в соответствующей структуре, чтобы запустить реша- тель;
* запущен решатель;
* получено решение или установлена неразрешимость задачи.

Результаты расчетов выводятся в отдельный файл с расширением .lst.

Получить оптимальное решение можно с помощью оператора *Display*, как это сделано в строке 58. В результате выполнения этого оператора будут напеча- таны значения переменных *x* и *z* в файле с расширением .lst после ключевого слова Execution.

Для запуска программы выберите меню *F ile* −−· *Run*. Появится вспомога-

тельное окно, в котором решатель сообщит о результатах компиляции и поиска решений. В случае успешного завершения работы решателя будет создан но- вый файл с расширением .lst, в котором появится полная информация о модели, процессе вычислений, статистике модели и результатах работы алгоритма.

Ниже приводятся результаты расчетов для рассмотренного примера.

\*\*\*\* REPORT SUMMARY : 0 NONOPT

0 INFEASIBLE

0 UNBOUNDED

G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m E x e c u t i o n

58 VARIABLE x.L сколько стульев $i$-го типа производит фирма 1 100.000, 2 72.000, 3 40.000, 4 53.000

58 VARIABLE z.L = 10294.000 суммарный доход от продажи стульев

127

58 EQUATION constraints1.L ограничения на кол-во деталей на складе

|  |  |  |
| --- | --- | --- |
| 1 1280.000, 2 1900.000, | 3 1060.000, 4 | 100.000, 5 165.000 |
| 6 760.000, 7 553.000, | 8 100.000, 9 | 72.000, 10 93.000 |

EXECUTION TIME = 0.000 SECONDS

Суммарный доход составляет 10294. Оптимальный план производства со- стоит из 100 стульев типа *капитан*, 72 стульев типа *помощник*, 40 стульев типа *маркиза* и 53 *испанских* стульев, остальные типы стульев в оптимальном решении отсутствуют. Дополнительно приводится информация о том, сколько деталей было задействовано при реализации оптимального плана. Эта инфор- мация может оказаться полезной при анализе решения.

Кроме упомянутых выше возможностей, в GAMS заложены и другие сред- ства для реализации сложных задач и взаимодействия с приложениями [18].

128

# 7. Список литературы

1. Алексеева Е. В., Кочетов Ю. А. Курс лекций по теории принятия реше- ний. URL: <http://www.math.nsc.ru/LBRT/k5/or.html>
2. Вентцель Е. С. Исследование операций. Задачи, принципы, методоло- гия. М.: КноРус, 2010.
3. Вершик А. М. Леонид Витальевич Канторович: человек и ученый: В 2 т. Новосибирск: Изд-во СО РАН. Филиал «Гео», 2002. Т. 1.
4. Гольштейн Е. Г., Юдин Д. Б. Задачи линейного программирования транс- портного типа. М.: Либроком, 2010.
5. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые зада- чи. М.: Мир, 1982.
6. Емеличев В. А., Ковалев М. М., Кравцов М. К. Многогранники графы, оптимизация (комбинаторная теория многогранников). М.: Наука, 1981.
7. Есипов Б. А. Методы исследования операций: Учебное пособие. СПб.: Лань, 2010.
8. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. Алгоритмы: по- строение и анализ. М.: Вильямс, 2009.
9. Кочетов Ю. А. Методы локального поиска для дискретных задач разме- щения Модели и алгоритмы. Saarbrucken: Lambert Academic Publishing, 2011.
10. Кочетов Ю. А., Плясунов А. В. Генетический локальный поиск для задачи о разбиении графа на доли ограниченной мощности // Журнал вычисли- тельной математики и математической физики. 2012. Т. 52. № 1.
11. Ларичев О. И. Теория и методы принятия решений, а также Хроника событий в Волшебных Странах: Учебник. М.: Логос, 2000.
12. Лэсдон Л. Оптимизация больших систем. М.: Наука, 1975.
13. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.
14. Ларин Р. М., Плясунов А. В., Пяткин А. В. Методы оптимизации. При- меры и задачи: Учеб. пособие, 2-е изд. Новосибирск, 2009.
15. Программное обеспечение для решения оптимизационных задач Advanced Integrated Mathematical Modeling System AIMMS. URL: [www.aimms.com](http://www.aimms.com/)
16. Программное обеспечение для решения оптимизационных задач AMPL Optimization. URL: [www.ampl.com](http://www.ampl.com/)

129

* 1. Программное обеспечение для решения оптимизационных задач IBM ILOG. URL: [www.ibm.com/software/websphere/products/optimization](http://www.ibm.com/software/websphere/products/optimization)
  2. Программное обеспечение для решения оптимизационных задач GAMS. URL: <http://gams.com/>
  3. Программное обеспечение для решения оптимизационных задач GNU Linear Programming Kit Package. URL: <http://www.gnu.org/software/> glpk/glpk.html
  4. Программное обеспечение для решения оптимизационных задач Microsoft Oﬃce, надстройка «Поиск решения» в Excel. URL: http://www.solver. com
  5. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программ- рование программирование: модели и вычислительные алгоритмы: Учеб. пособие. М.: ФИЗМАТЛИТ, 2007.
  6. Схрайвер A. Теория линейного и целочисленного программирования: В 2 т. М.: Мир, 1991.
  7. Appa G. M., Pitsoulis L. S., Paul W. H. (Eds.) Handbook on Modelling for Discrete Optimization. Springer Series Vol. International Series in Operations Research & Management Science 2006. V. 88, XXII.
  8. Eppen G. D., Gould F. J., Schmidt C. P. Introductory management science.

4*th* Edition. A Simon & Schuster Company. 1993.

* 1. Pochet Y., Wolsey L. A. Production planning by mixed integer programming. In: T. V. Mikosh, S. I. Resnick, S. M. Robinson (Eds.) Springer Series in Operations Research & Financial Engineering. 2006.
  2. Plastria F. Formulating logical implications in combinatorial optimization // European journal of Operational Research. 2002. № 140.
  3. Ehrgott M. Multicriteria Optimization. 2*nd* Edition. Berlin: Springer, 2005.
  4. Eiselt H. A., Sandblom C.–L. Operations Research: A Model-Based Approach. Berlin, Heidelberg: Springer-Verlag, 2010.
  5. Kellerer H., Pferschy U., Pisinger D. Knapsack problems. Berlin, Heidelberg: Springer-Verlag, 2004.
  6. Nemhauser G. L., Wolsey L. A. Integer and combinatorial optimization. John Wiley & Sons, Inc., 1999.
  7. Williams H. P. Model Building in Mathematical Programming. 4*th* Edition, University of Southampton, John Wiley & Sons, Inc., 1999.

130

Учебное издание

**Алексеева** Екатерина Вячеславовна

## ПОСТРОЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ. ПРИМЕРЫ И ЗАДАЧИ

Учебное пособие

В оформлении обложки использована иллюстрация из детского

анимационного сериала «Фиксики». Все права на объекты, входящие в состав указанного анимационного сериала, такие как персонажи, музыкальные

произведения, принадлежат закрытому акционерному обществу «Аэроплан».

Редактор *Е. П. Войтенко*

Подписано в печать 26.03.2012г.

Формат 60 84 1/16. Печать офсетная.

×

Уч.-изд.л. 8. Усл. печ. л. 7,44. Тираж 200 экз.

Заказ №

Редакционно-издательский центр НГУ. 630090, Новосибирск - 90, ул. Пирогова, 2.