Задача об оптимальном плане выпуска стульев

Суть задачи такова. Мебельная фабрика может выпускать стулья двух типов ценою в 8 и 12 условных единиц (у. е.[1]). Под этот заказ выделены материальные и людские ресурсы — известно, сколько досок, ткани и времени идет на изготовление каждого стула (табл. 4.1).

Таблица 4.1. *Данные для задачи об оптимальном плане выпуска стульев*

Стул	**Расход досок, м**	**Расход ткани, м^2**	**Расход времени, человеко-часов**
Первый	2	0.5	2
Второй	4	0.25	2.5
Ресурс	490	65	320

Спрашивается, как нужно спланировать производство стульев, чтобы наделать их либо количеством, либо ценой поболее. Решение этой задачи представлено на рис. 1 и 2.

Данная задача относится к широкому классу задач под названием *"задачи линейного программирования"*: необходимо установить *план* (программу!) выпуска изделий (у нас это стулья), ориентируясь на *целевую функцию* (у нас их две — общее количество и общая стоимость стульев) и принимая во внимание *ограничения* (ресурсы по доскам, ткани и человеко-часам).

> ***Примечание***
> "Линейного" — значит, что и целевая функция, и ограничения от переменных задачи зависят линейно. Слово "программирование" не имеет прямого отношения к программированию в современном понимании этого слова. Здесь другой смысл — программа (план) выпуска продукции. Задачу приходилось решать задолго до появления компьютеров.

На рис. 1 показана попытка решения задачи о стульях с помощью функции `Maximize`. Она оказалась не вполне удачной: Mathcad, а точнее, функции `Maximize` и `Minimize` в стандартной их постановке *(см. далее)* не способны решать целочисленные задачи, т. е. такие, где в списке ограничений стоит ограничение на целочисленность искомых переменных.

---

[1] Это, естественно, не доллары США, а на самом деле *условные единицы*, не влияющие на решение задачи. Хотя единицу измерения стоимости — доллар — мы в расчет ввели.

▼ Пользовательские единицы

$шт := mole \quad пм := m \quad м := m \quad чел\text{-}час := шт \cdot hr$

▲ Пользовательские единицы

Задача о плане выпуска стульев двух моделей

Первая целевая функция $Количество(Стул_1, Стул_2) := Стул_1 + Стул_2$

Вторая целевая функция $Цена(Стул_1, Стул_2) := 8\frac{\$}{шт}\cdot Стул_1 + 12\frac{\$}{шт}\cdot Стул_2$

Начальное приближение $Стул_1 := 1\,шт \quad Стул_2 := 1\,шт$

Решение по первой целевой функции

Given Ограничения по ресурсам: доски, обивочная ткань и

доски $2\frac{пм}{шт}\cdot Стул_1 + 4\frac{пм}{шт}\cdot Стул_2 \le 490\,пм$

обивочная ткань $0.5\frac{м^2}{шт}\cdot Стул_1 + 0.25\frac{м^2}{шт}\cdot Стул_2 \le 65\,м^2$

рабочая сила $2\frac{чел\text{-}час}{шт}\cdot Стул_1 + 2.5\frac{чел\text{-}час}{шт}\cdot Стул_2 \le 320\ чел\text{-}час$

$$\begin{pmatrix} Стул_1 \\ Стул_2 \end{pmatrix} := Maximize(Цена, Стул_1, Стул_2)$$

$$\begin{pmatrix} Стул_1 \\ Стул_2 \end{pmatrix} = \begin{pmatrix} 18.33 \\ 113.33 \end{pmatrix} шт$$

$Количество(Стул_1, Стул_2) = 131.67\,шт$

$Цена(Стул_1, Стул_2) = 1506.67\,\$$

Рис. 1. Попытка решения задачи целочисленного линейного программирования

На рис. 2 сделана попытка спасти решение, показанное на рис. 1. В список ограничений введено ограничение на целочисленность: `Стул₁ = Floor(Стул₁,шт)` — переменная `Стул₁` должна быть равна своей целой части. После этой вставки решение было выдано целочисленное (1 и 122 стула)[2]. Но самое ли оно оптимальное (стоимость стульев равна 1472 $)? Ответ на этот вопрос хранится на рис. 2.

---

[2] Нужно всегда пытаться подсовывать инструментам Mathcad необычные сочетания исходных данных. Хотя бы из-за любопытства узнать, как на это программа среагирует. Но использовать дальше полученные таким образом ответы нужно очень осторожно, т. к. за недокументированные приемы ("нарушение правил эксплуатации") фирма-разработчик ответственности не несет.

$$\text{Стул}_1 = \text{Floor}(\text{Стул}_1, \text{шт})$$

$$\begin{pmatrix} \text{Стул}_1 \\ \text{Стул}_2 \end{pmatrix} := \text{Maximize}(\text{Цена}, \text{Стул}_1, \text{Стул}_2)$$

$$\begin{pmatrix} \text{Стул}_1 \\ \text{Стул}_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 122 \end{pmatrix} \text{шт} \qquad \begin{array}{l} \text{Количество}(\text{Стул}_1, \text{Стул}_2) = 123\,\text{шт} \\ \text{Цена}(\text{Стул}_1, \text{Стул}_2) = 1472\,\$ \end{array}$$

Рис. 2. Попытка спасения решения задачи целочисленного линейного программирования

На рис. 3 показано решение задачи о плане (программе) выпуска стульев методом перебора — анализа матрицы `Цена`, хранящей стоимости вариантов выпуска стульев при выполнении ограничений — если одно из ограничений не соблюдается, то соответствующий элемент матрицы `Цена` становиться нулевым. Встроенная в Mathcad функция `match` в задаче о стульях вернула нам два решения задачи, при которой цена стульев (20 и 112 шт.; 17 и 114 шт.) будет максимальной — 1504 $. Так что решение, показанное на рис. 1 (1 и 122 стульев), было не оптимальным.

Пользовательские единицы

$шт := 1 \quad пм := m \quad м := m \quad чел\text{-}час := шт \cdot hr$

Пользовательские единицы

Задача о плане выпуска стульев двух моделей

Первая целевая функция $\quad Количество(Стул_1, Стул_2) := Стул_1 + Стул_2$

Вторая целевая функция $\quad Цена(Стул_1, Стул_2) := 8\frac{\$}{шт} \cdot Стул_1 + 12\frac{\$}{шт} \cdot Стул_2$

$Стул_1 := 0шт..150шт \qquad Стул_2 := 0шт..150шт$

$$Цена_{Стул_1, Стул_2} := \left| \begin{array}{l} 0\$ \text{ if } 2\frac{пм}{шт} \cdot Стул_1 + 4\frac{пм}{шт} \cdot Стул_2 > 490\,пм \\ 0\$ \text{ if } 0.5\frac{м^2}{шт} \cdot Стул_1 + 0.25\frac{м^2}{шт} \cdot Стул_2 > 65\,м^2 \\ 0\$ \text{ if } 2\frac{чел\text{-}час}{шт} \cdot Стул_1 + 2.5\frac{чел\text{-}час}{шт} \cdot Стул_2 > 320\ чел\text{-}час \\ Цена(Стул_1, Стул_2) \text{ otherwise} \end{array} \right.$$

$Max\_Цена := max(Цена) \qquad Max\_Цена = 1504\,\$$

$Стул := match(Max\_Цена, Цена) \qquad Стул^T = \left[\begin{pmatrix} 20 \\ 112 \end{pmatrix} \ \begin{pmatrix} 17 \\ 114 \end{pmatrix} \right]$

$\begin{pmatrix} Стул_1 \\ Стул_2 \end{pmatrix} := Стул_0 \qquad Количество(Стул_1, Стул_2) = 132\,шт$

$2\frac{пм}{шт} \cdot Стул_1 + 4\frac{пм}{шт} \cdot Стул_2 = 488\,пм \qquad 0.5\frac{м^2}{шт} \cdot Стул_1 + 0.25\frac{м^2}{шт} \cdot Стул_2 = 38\,м^2$

$2\frac{чел\text{-}час}{шт} \cdot Стул_1 + 2.5\frac{чел\text{-}час}{шт} \cdot Стул_2 = 320\ чел\text{-}час$

$\begin{pmatrix} Стул_1 \\ Стул_2 \end{pmatrix} := Стул_1 \qquad Количество(Стул_1, Стул_2) = 131\,шт$

$2\frac{пм}{шт} \cdot Стул_1 + 4\frac{пм}{шт} \cdot Стул_2 = 490\,пм \qquad 0.5\frac{м^2}{шт} \cdot Стул_1 + 0.25\frac{м^2}{шт} \cdot Стул_2 = 37\,м^2$

$2\frac{чел\text{-}час}{шт} \cdot Стул_1 + 2.5\frac{чел\text{-}час}{шт} \cdot Стул_2 = 319\ чел\text{-}час$

Рис. 3. Решение задачи целочисленного линейного программирования перебором вариантов

Внизу рис. 3 можно увидеть, как расходуются ресурсы (что остается лишним — доски, ткань и/или рабочее время) при найденных двух планах — производственных программах изготовления стульев.

На рис. 4.26 можно видеть решение задачи о стульях в случае, когда к пакету Mathcad подгружено расширение SOEP (Solving Optimization Extension Pack) и когда в списке аргументов функции `Maximize` можно приписать дополнительный аргумент "II", означающий, что переменные $Стул_1$ и $Стул_2$ должны быть целочисленными — I, integer.

Given Ограничения по ресурсам: доски, обивочная ткань и рабочая сила

$$2\frac{пм}{шт}\cdot Стул_1 + 4\frac{пм}{шт}\cdot Стул_2 \le 490\,пм$$

$$0.5\frac{м^2}{шт}\cdot Стул_1 + 0.25\frac{м^2}{шт}\cdot Стул_2 \le 65\,м^2$$

$$2\,\frac{чел\text{-}час}{шт}\cdot Стул_1 + 2.5\,\frac{чел\text{-}час}{шт}\cdot Стул_2 \le 320\ чел\text{-}час$$

$$\begin{pmatrix} Стул_1 \\ Стул_2 \end{pmatrix} := \mathrm{Maximize}(Цена,\ Стул_1,\ Стул_2, \text{"II"})$$

$$\begin{pmatrix} Стул_1 \\ Стул_2 \end{pmatrix} = \begin{pmatrix} 20 \\ 112 \end{pmatrix} шт \qquad Количество(Стул_1, Стул_2) = 132\,шт$$

$$Цена(Стул_1, Стул_2) = 1504\,\$$$

Рис. 4.26. Решение задачи целочисленного линейного программирования в среде Mathcad с помощью пакета расширения SOEP

Задача, показанная на рис. 1, 2 и 3 простенькие, но очень, если так можно выразиться, жизненно важные. На каждом шагу приходится что-то оптимизировать (расходы, например), принимая во внимание всякого рода ограничения (доходы!). Можно привести такой пример. После часа пик (скажем, в зимнее утро) расход электроэнергии падает, и необходимо снижать нагрузку электрогенераторов электростанций. Как это делать? Можно отключить отдельные генераторы, а можно оставить их в работе, изменив нагрузку. Диспетчер энергосистемы дает соответствующие команды, ориентируясь на некие целевые функции: средний расход топлива по системе, выброс с дымовыми газами вредных веществ в атмосферу, износ оборудования, степень готовности электростанций и дальше менять нагрузку и т. д. Переменные такой оптимизации могут быть и вещественными (мощность отдельного энергоблока, которая меняется, естественно, в разумных пределах, определяемых техническими условиями — ограничения в задаче), и целочисленными (количество работающих блоков). Эта задача очень сложная, но и весьма эффективная — здесь речь идет о высвобождаемых составах с топливом, о снижении выбросов CO_2 в атмосферу (вспомним Киотский протокол) и т. д.

Вот еще примеры. Когда нужно убирать пшеницу? Пораньше — зерно еще не вызрело. Попозже — часть зерна уже осыпалась. Сколько и каких акций стоит купить на ограниченную сумму денег, чтобы будущий дивиденд был максимален? В каких средствах массовой информации стоит размещать рекламу на выделенные по смете деньги, чтобы эффект от нее был максимален и т. д. и т. п.?