ЭКОНОМИКА И МАТЕМАТИЧЕСКИЕ МЕТОДЫ
1965 том I, вып. 2

ОПТИМАЛЬНЫЙ РАСКРОЙ ПОЛОСЫ

В. Г. МЕДНИЦКИЙ

(*МОСКВА*)

Во многих производствах возникает проблема рационального раскроя полосы на прямоугольники (пластины) определенных размеров. Эта задача является частным случаем более общей задачи рационального раскроя материалов, математическая постановка которой была дана Л. В. Канторовичем [1]. При ее решении возникают трудности двоякого рода. Во-первых, сведение к определенной математической схеме (например, линейного программирования) может быть выполнено после описания вариантов раскроя при помощи соответствующим образом подобранных многомерных величин. Как правило, нельзя указать регулярного способа образования этих величин.

Вторая трудность состоит в том, что даже при небольшой размерности вектора, описывающего различные варианты раскроя, его составляющие получаются при помощи комбинаторного перебора. Поэтому число различных возможных вариантов раскроя чрезвычайно велико и решение соответствующей математической задачи связано с переработкой и хранением огромных количеств информации. Возникающие вычислительные трудности настолько значительны, что решение задачи в сколько-нибудь приемлемое время оказывается невозможным.

Описываемый ниже метод основывается на введении в задачу дополнительного нелинейного ограничения, описывающего допустимые варианты раскроя. Получающийся в результате вычислительный процесс является по существу модификацией одного из методов решения общей задачи программирования, который описан Ф. Вольфом и Дж. Данцигом [2]. Объем исходной и промежуточной информации существенно сокращается, и решение задачи оказывается в пределах возможностей ЭВМ малой мощности типа «Урал-1», «Минск-1» и т. п.

Постановка задачи. Исходным материалом для производства некоторых изделий промышленности служат прямоугольные пластины, которые выкраиваются из стандартных рулонов. Процесс раскроя протекает следующим образом. Пластины систематизируются по меньшему размеру, а затем определяется общая длина полосы, которую образуют все пластины с данным наименьшим размером. Рулон разрезается в направлении наибольшего размера на полосы, ширина которых равна ширине пластин нужного типа. Затем полосы режутся на пластины. При этом вдоль рулона образуется полоса материала, которая дальше использована быть не может и идет в отход (рис. 1).

Задача легко может быть описана в схеме линейного программирования введением участков однородного раскроя. Под участком однородного раскроя мы понимаем часть рулона длины l_j, на которой располагается p_1 полос ширины Δ_1, p_2 — ширины Δ_2 и т. д. Пусть имеется всего N участков однородного раскроя, тогда общая длина полосы шириной Δ_i, полученной таким способом, будет равна $\sum_j l_j p_{ij}$, где p_{ij} — число полос ширины Δ_i на участке l_j, и, следовательно, задача программирования ставится следующим образом:

$$\min \sum_{j=1}^{N} \varepsilon_j l_j \tag{1}$$

при

$$\sum_{j=1}^{N} p_{ij} l_j = a_i, \quad l_j \geqslant 0.$$

Здесь ε_j — ширина полосы отходов на j-м участке и a_i — необходимая общая длина полосы с шириной Δ_i (в дальнейшем для краткости i-го типа). После раскроя полосы на пластины в каждой полосе получается дополнительный отход, так как длина пластины может не укладываться целое число раз в длину участка однородного раскроя. Однако общая площадь отходов второго типа значительно меньше, чем площадь основной полосы отходов, получающейся при раскрое рулона по длине. На этом основании отходами второго типа можно пренебречь. Величины $\{p_{ij}\}$, составляющие матрицу ограничений задачи, должны быть определенным набором целых чисел. Кроме того, очевидно, что допустимый набор величин ε_j, p_{1j}, ..., p_{mj}, который мы в дальнейшем будем называть вариантом раскроя, должен удовлетворять следующей системе ограничений:

$$\sum_{i=1}^{m} p_{ij}\Delta_i + \varepsilon_j = \Delta, \quad \varepsilon_j \geqslant 0, \tag{2}$$

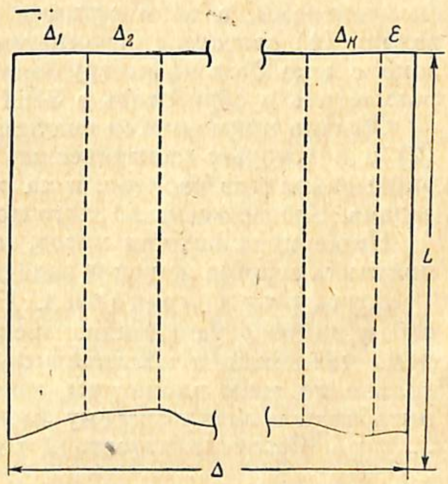

Рис. 1. Схематическое изображение полосы раскроя: Δ — общая ширина рулона; Δ_i — ширина полосы i-го типа; ε — ширина полосы отходов; L — длина рулона

так как в ней по существу утверждается, что ширина отхода плюс ширина всех участвующих в варианте раскроя полос равна общей ширине рулона.

Любопытно, что допустима эквивалентная постановка (1) в виде

$$\min \sum_{j=1}^{N} l_j \tag{3}$$

при

$$\sum_{j=1}^{N} p_{ij} l_j = a_i, \quad l_j \geqslant 0.$$

В этом варианте мы стремимся «выпустить» необходимый набор полос при максимальном уменьшении общей длины раскраиваемого полотна. Физически очевидно, что этого можно добиться только за счет раскроя с минимальным отходом. Действительно, $\varepsilon_j = \Delta - \sum_{i=1}^{m} p_{ij}\Delta_i$ в силу соотношения (2). Подставляя в функционал задачи (1), получаем

$$\sum \varepsilon_j l_j = \Delta \sum l_j - \sum_j l_j \sum_{i=1}^{m} p_{ij}\Delta_i =$$

$$= \Delta \sum l_j - \sum_{i=1}^{m} \Delta_i \sum_j p_{ij} l_j = \Delta \sum l_j - \sum_{i=1}^{m} \Delta_i a_i.$$

Таким образом, функционал (1) является положительным линейным преобразованием функционала (3) и, следовательно, (1) и (3) эквивалентны.

Число различных по размерам полос в возникающих практических задачах m очень невелико и обычно не превышает двух-трех десятков. В то же время число возможных вариантов раскроя огромно и исчисляется десятками, а то и сотнями тысяч. Поэтому уже получение матрицы задачи (1) связано с огромными трудностями, а решение этой задачи связано с еще большими трудностями, возникающими из-за необходимости разместить и обработать в ЭВМ огромное количество информации.

Обычно применяется полуэмпирический метод, при котором с помощью (2) и некоторых эмпирических критериев строится небольшая часть матрицы вариантов раскроя, и на полученной выборке решается симплексная задача. Несовершенство этого метода очевидно.

Ниже предлагается метод, который позволяет обойти эти трудности и получить строгое решение задачи.

Алгоритм решения. Предположим, что удалось каким-либо способом найти ограниченное число вариантов раскроя, среди которых заведомо находятся и оптимальные. Тогда, решая задачу программирования только по этим вариантам, можно точно выделить оптимальные, определить план и найти систему оценок. Обозначим набор оценок как $u_1, u_2, \ldots, u_m$. Согласно известной теореме двойственности относительно способов, вошедших в план, будет выполняться равенство $\sum_{i=1}^{m} u_i p_{ij} - \varepsilon_j = 0$, относительно прочих — неравенство

$$\sum_{i=1}^{m} u_i p_{ij} - \varepsilon_j < 0. \tag{a}$$

Предположим, что мы хотим проверить правильность выбора исходных способов. Для этого надо просмотреть все остальные и убедиться, что по отношению к ним неравенство (a) также выполнено. Пусть это неверно и нашелся k-й способ, для которого $\Sigma u_i p_{ik} - \varepsilon_k > 0$. Тогда «оптимальный» план может быть улучшен введением этого способа в первоначально отобранную группу. Решением расширенной таким образом задачи находят новый план и новую систему оценок. Затем процесс может быть повторен. Этот прием не упрощает решения, так как каждый цикл включает просмотр большого числа неравенств, но приводит к соображениям, которые позволяют в конечном счете решить задачу.

Действительно, возникает вопрос, нельзя ли, зная систему оценок, рассчитанную по ограниченному числу способов, найти, не просматривая матрицу способов, тот, который улучшает план, или доказать, что любой способ не позволяет улучшить план и, следовательно, оптимум найден. Критерий, отвечающий на этот вопрос, легко указать. Действительно, если $\max\left\{\sum u_i p_{ij} - \varepsilon_j\right\}$ при всех возможных p_{ij} и ε_j не положителен, то не существует варианта раскроя, улучшающего найденный план. Легко сформулировать, что́ следует понимать под любым возможным набором $\{p_{ij}, \varepsilon_j\}$.

Эти величины должны удовлетворять уже приведенным условиям (2). Таким образом, задача (1) может быть решена так.

1. По некоторому числу способов найти оптимальное решение и двойственные оценки задачи:

$$\min \sum_{j=1}^{m} \varepsilon_j l_j$$

при ограничениях

$$\sum_{j=1}^{m} p_{ij} l_j = a_i, \quad l_j \geqslant 0, \tag{4}$$

n — здесь невелико, например, $n = m$.

2. С найденными оценками $u_1, u_2, \ldots, u_m$ построить задачу целочисленного программирования

$$\max \left\{ \sum_{i=1}^{m} p_i u_i - \varepsilon \right\}$$

при условии целочисленности переменных $p_1, p_2, \ldots, p_m$ и ограничениях

$$\sum_{i=1} p_i \Delta_i + \varepsilon = \Delta, \quad \varepsilon \geqslant 0 \tag{5}$$

и найти ее решение.

3. Если максимум в задаче (5) положителен, то присоединить набор p_i, ε к условиям (4) и повторить пункты 1 и 2. Если максимум равен нулю, то оптимальное решение найдено в пункте 1.

Процесс по-прежнему циклический, но мы избавились от просмотра на каждой итерации большого числа неравенств, получив взамен задачу целочисленного программирования. Ее решение общими методами трудоемко и в нашем случае может быть упрощено специальным приемом, который состоит в следующем.

Введем в рассмотрение функции произвольного аргумента $f_i(y_i)$, определенные следующим образом:

$$f_i(y_i) = \max \{p_i u_i - \varepsilon_i\},$$

где $p_i \Delta_i + \varepsilon_i = y_i$ и p_i — целые, $\varepsilon_i \geqslant 0$.

Тогда задача (5) эквивалентна следующей:

$$\max \sum_{i=1}^{m} f_i(y_i)$$

при ограничениях

$$\sum y_i = \Delta, \quad y_i \geqslant 0. \tag{6}$$

Последняя просто решается методами динамического программирования. Функция $f_i(y_i)$ легко строится для отрезка $0 \leqslant y \leqslant \Delta$, как только определены оценки, именно она линейна и непрерывна слева на отрезках $[(k-1)\Delta_i \leqslant x < k\Delta_i]$. Прямолинейный отрезок функции параллелен биссектрисе четвертого квадранта, и в точках $k\Delta_i$ функция претерпевает скачок h_i, равный $\Delta_i + u_i$, как это изображено на рис. 2.

Остановимся на смысле проделанных преобразований. Довольно ясно, что возможен любой вариант раскроя, удовлетворяющий ограничению $\sum p_i \Delta_i + \varepsilon = \Delta$ и условию целочисленности $\{p_i\}$, так как этой системой ограничений по существу описывается техника раскроя. Можно считать, что Δ_i, Δ выражены, например, в миллиметрах и являются целыми числами. Тогда ε тоже должно быть целым, а равенство заведомо имеет допустимые решения в целых числах. При любом варианте раскроя желательно получить больше полос определенных типов с минимальным отходом. Эти два требования необходимо совместить в одном — максимизации или минимизации какого-то синтетического показателя. Такой показатель можно найти разными способами, например, в виде условия максимальности $\{\sum \lambda_i p_i - \varepsilon\}$, где λ_i — некоторые веса. Если все веса положительны, то максимизация может осуществляться как за счет роста величин $\{p_i\}$, так и благодаря уменьшению ε, что и требуется по смыслу задачи. Веса должны оценивать дефицитность полос различных типов в промежуточном плане. Естественно поэтому, что весами оказываются двойственные оценки этих планов. В результате мы получили вместо матрицы ограничений задачу целочисленного программирования, которая, однако, в данном случае играет ту же роль — роль простого хранителя информации. Разница заключается в том, что вместо в с е й и н ф о р м а ц и и машина должна запомнить лишь способ получения л ю б о й н е о б х о д и м о й и н ф о р м а ц и и, а при обращении к накопителю каждый раз нужно точно сформулировать, какая именно информация в данный момент требуется (критерий оптимальности!). Это позволяет каждый раз осуществлять выбор либо однозначно, либо в виде равноценных альтернатив. Такая, если можно употребить это выражение, динамическая память позволяет резко сократить объем запоминаемой информации. Вместо нескольких миллионов элементов, составляющих матрицу ограничений, оказывается достаточным запомнить максимум 200—300 команд программы счета экстремальной задачи. Процесс выбора осуществляется более экономно. При нахождении оптимального варианта просматривается только небольшая часть остальных, большинство же отбрасывается сразу.

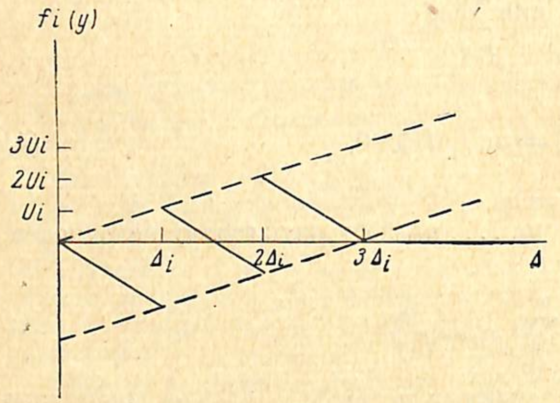

Рис. 2. Схематическое изображение функции $f_i(y_i)$

ОБ ЭКВИВАЛЕНТНОСТИ ЗАДАЧ ЦЕЛОЧИСЛЕННОГО ПРОГРАММИРОВАНИЯ (5) И ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ (6)

Как было показано выше, основная задача раскроя (1) допускает эквивалентную постановку в форме (2). При этом вспомогательную задачу целочисленного программирования можно переписать в виде

$$\max \sum p_i u_i \qquad (7)$$

при условии $\sum p_i \Delta_i \leqslant \Delta$ и целочисленности переменных $(p_1, p_2, \ldots, p_m)$.

Вариант в этом случае будет полноценным, если значение функционала (7) окажется большим единицы.

Функции в соответствующей задаче динамического программирования при этом переопределяются следующим образом: $\bar{f}_i(y) = \max\limits_{p_i} \{p_i u_i\}$ при $p_i \Delta_i \leqslant y$ и p_i — целом.

Очевидно, что максимум последнего выражения достигается при максимально возможном значении p_i, и, следовательно, $\bar{f}_i(y_i)$ можно определить как

$$\bar{f}_i(y) = u_i\left[\frac{y}{\Delta_i}\right]. \tag{8'}$$

Это кусочно-ступенчатая функция, не меняющая своих значений на отрезках $[k\Delta_i, (k+1)\Delta_i]$. В точках, кратных Δ_i, она претерпевает скачок u_i.

Задача динамического программирования после указанных преобразований принимает вид:

$$\max \sum \bar{f}_i(y_i) \quad \text{при} \quad \sum y_i \leqslant \Delta, \quad y_i \geqslant 0. \tag{8}$$

Как известно, решение (8) по методу динамического программирования сводится к построению двух рекуррентных последовательностей:

1) $F_1(z) = \bar{f}_1(z)$,

$$F_{k+1}(z) = \max_{0 \leqslant y \leqslant z} \{f_{k+1}(y) + F_k(z-y)\}, \tag{9}$$

$$z \in [0, \Delta];$$

2) z_m из равенства: $F_m(z_m) = \max\limits_{0 \leqslant z \leqslant \Delta} F_m(z)$,

$$\hat{y}_m = y_m(z_m), \tag{10}$$

и z_{m-1} из равенства $F_{m-1}(z_{m-1}) = \max\limits_{0 \leqslant z \leqslant z_m - \hat{y}_m} \{F_{m-1}(z)\}$, где $y_{k+1}(z)$ — точка, в которой достигается максимум второго выражения из (9). Последовательность $\{\hat{y}_m, \hat{y}_{m-1}, \ldots, \hat{y}_1\}$ и дает решение (оптимальное поведение в терминологии Р. Беллмана). Основой доказательства эквивалентности (7) и (8) является лемма 1.

Лемма 1. *Пусть* $F(z) = \max\limits_{0 \leqslant y \leqslant z} [f(y) + F'(z-y)]$, *где* $F'(u)$ — *произвольная, монотонно неубывающая функция аргумента. Тогда* $F(z)$ *является монотонно неубывающей функцией* z.

По определению $F(z_1) = f(\bar{y}) + F'(z_1 - \bar{y})$, где z_1 — произвольно и $\bar{y} \subset [0, z_1]$. Пусть $z_2 \geqslant z_1$, тогда в силу монотонности $F'(z_2 - \bar{y}) \geqslant F'(z_1 - \bar{y})$, и, следовательно, $F(z_1) \leqslant f(\bar{y}) + F'(z_2 - \bar{y})$. С другой стороны, $F(z_2) = \max\limits_{0 \leqslant y \leqslant z,} [f(y) + F'(z-y)]$ и, значит, $f(\bar{y}) + F'(z_2 - \bar{y}) \leqslant F(z_2)$.

Таким образом, $F(z_1) \leqslant F(z_2)$, если только $z_1 \leqslant z_2$, что и завершает доказательство.

Следствие 1. Пусть $F(z) = \max \sum \varphi_i(x_i)$ при $\sum x_i = z$, $x_i \geqslant 0$, $z \in [a, b]$.

Если среди $\{\varphi_i(x)\}$ найдется монотонно неубывающая функция на $[a, b]$, то $F(z)$ на $[a, b]$ — монотонно неубывающая. Действительно, пусть монотонно не убывает $\varphi_1(z)$. Выберем $F_1(z) = \varphi_1(z)$, тогда каждая из функций обладает требуемым свойством.

Следствие 2. Последовательность (10) может быть преобразована к виду

$$z_m = \Delta, \quad \hat{y}_m = y_m(\Delta), \; z_{m-1} = \Delta - \hat{y}_m \ldots \tag{10'}$$

Пусть $\bar{f}(y)$ принадлежат основной системе функций задачи динамического программирования и $F(u)$ — кусочно-ступенчатая неубывающая функция аргумента.

Лемма 2. *Пусть* ξ_1 *и* ξ_2 *таковы, что* ξ_1 *и* $z - \xi_2$ — *соответственно ближайшие точки разрыва функций* f *и* F. *Тогда:*

1) *значение* $\max\limits_{0 \leqslant y \leqslant z} \{f(y) + F(z-y)\}$ *между точками разрыва не меняется;*

2) *отрезки* (ξ_2, ξ_1), (ξ_1, ξ_2) *при* $\xi_1 \geqslant \xi_2$ *и* $\xi_1 \leqslant \xi_2$ *соответственно являются множеством точек локального минимума (максимума) суммы* $\psi(z) = f(\xi) + F(z - \xi)$.

Доказательство. Первое утверждение очевидно. Для доказательства второго заметим, что при $\xi_1 < \xi_2$ можно найти такое ε, что $F(z - \xi_1 + \varepsilon)$ будет равно $F(z - \xi_1)$, в то время как $f(\xi_1 - \varepsilon) < f(\xi_1 + 0)$. Следовательно, значение суммы при $\xi_1 - \varepsilon$ меньше, чем при $\xi_1 + 0$. Проводя аналогичные рассуждения относительно точки ξ_2, убеждаемся, что отрезок $[\xi_1\xi_2]$ есть область локального максимума суммы. Доказательство в случае $\xi_2 < \xi_1$ проводится аналогично.

Интервалы рассмотренного выше типа покрывают весь отрезок $[0, \Delta]$. Максимального значения сумма может достигнуть только на таком интервале, у которого левый конец является точкой разрыва f и правый — функции F.

Таким образом, область оптимальных значений y может состоять из некоторого количества интервалов и изолированных точек.

Пусть $\{[\xi_1^{(p)}, \xi_2^{(p)}]\}$ является множеством оптимальных поведений в точке z и $\max\limits_p (\xi_2^p - \xi_1^p) = l = (\xi_2^{\bar p} - \xi_1^{\bar p})$. Легко доказать, что, если выбрать в качестве оптимального поведения точку $\xi_1^{\bar p}$, то значения $\varphi(z) = \max\limits_{0 \leqslant y \leqslant z} \{f(y) + F(z - y)\}$ сохраняются на отрезке $[z - l, z]$.

Действительно, так как $\xi_1^{(\bar p)}$ — оптимальное поведение в точке z, то

$(\varphi(z) = \max \{f(\xi) + F(z - \xi)\} = f(\xi_1^{\bar p}) + F(z - \xi_1^{\bar p})$, но по непрерывности $F(z - \xi_1^{\bar p}) = F(z - \xi_2^{\bar p} + 0) = F[z - (\xi_2^{\bar p} - \xi_1^{\bar p})] = F(z - l - \xi_1^{\bar p})$, откуда $\max\limits_{\leqslant \xi \leqslant z} \{f(\xi) + F(z - \xi)\} = F[(z - l) - \xi_1^{\bar p}] + f(\xi_1^{\bar p})$.

В силу монотонности $\varphi(z)$ заключаем, что $\xi_1^{\bar p}$ является оптимальным поведением на отрезке $[z - l, z]$, а $\varphi(z)$ на данном отрезке постоянна. Легко доказать, что $\varphi(z - l - \varepsilon) < \varphi(z)$ при $\varepsilon > 0$.

Предположим, что существует оптимальное поведение такое, что

$$f(u) + F(z - l - \varepsilon - u) = \varphi(z). \tag{11}$$

В этом случае любая точка отрезка $[u, u + l + \varepsilon]$ должна определять оптимальное поведение в точке z, так как в силу монотонности при $u \leqslant u' \leqslant u + l + \varepsilon$, $f(u') \geqslant f(u)$ и $F(z - u') \geqslant F(z - l - \varepsilon - u)$, а следовательно, в точке z существует отрезок оптимальных поведений длины $l + \varepsilon$. Однако l по построению максимально. В силу этого противоречия равенство (11) выполнено быть не может и, значит, $\varphi(z - \varepsilon - l) < \varphi(z - l)$.

Таким образом, может быть сформулирована теорема: *если* $f(y)$ *принадлежит основной системе* (8') *и* $F(z)$ — *ступенчатая, неубывающая функция, то* $\varphi(z) = \max \{\bar f(y) + F(z - y)\}$ *является ступенчатой, неубывающей функцией* z; *в качестве оптимального поведения всегда может быть выбрана одна из точек разрыва* $\bar f(y)$.

Этим доказано, что среди множества оптимальных поведений (8) содержится решение задачи (7).

ЛИТЕРАТУРА

1. Л. В. Канторович. Математические методы организации и планирования производства. Л., изд-во ЛГУ, 1939.
2. G. B. Dantzig, Ph. Wolfe. Decomposition principle for linear programs. Opns Res., 1960, vol. 8, No. 1.

Поступила в редакцию
23 XI 1964