

УДК 330.45

## УПРАВЛЕНИЕ ИНТЕГРАЦИЕЙ ИННОВАЦИОННЫХ ПРОЕКТОВ В ПРЕДМЕТНУЮ ОБЛАСТЬ

**Д. Б. Босов,**

канд. экон. наук, глава представительства  
Компания «Алтек инвестментс лимитед»

Предложены модель и алгоритм решения задачи оптимизации многомерных параметрических рядов в интересах рациональной интеграции инновационных проектов в предметную область.

**Ключевые слова** — управление, инновационный проект, математическая модель, множество, вариант математической формализации, решение задачи.

### Постановка задачи

Любой инновационный проект создается и реализуется в интересах удовлетворения некоторого спектра характерных для соответствующей предметной области его использования потребностей. Поскольку все они, как правило, не могут быть полностью обеспечены каким-либо одним вариантом инновационного продукта, рациональная интеграция инновационных проектов в предметную область путем формирования соответствующих параметрических рядов инновационных продуктов (изделий) становится актуальной задачей инновационного менеджмента.

Сложность современных инновационных проектов и разнообразие предъявляемых к ним требований приводят к тому, что проблемы формирования таких рядов в процессе инновационного развития становятся весьма непростыми. Попытки решить их на основе предыдущего опыта и интуиции часто приводят к грубым ошибкам, что в условиях высокой стоимости и большой длительности разработки и производства инноваций недопустимо. Поэтому решение указанных проблем должно опираться также и на объективные научные обоснования. Инструментом для таких обоснований могут быть соответствующие математические модели. В настоящей работе предложены модель и алгоритм решения задачи оптимизации многомерных параметрических рядов в интересах рациональной интеграции инновационных проектов в предметную область.

Сущность задачи состоит в следующем.

Задано множество  $J = \{1, 2, \dots, j, \dots, N\}$  видов потребностей. Каждый вид потребности опреде-

ляется совокупностью многих характеристик (показателей, параметров).

Имеется также множество  $I = \{1, 2, \dots, i, \dots, M\}$  изделий, способных совместно обеспечить все виды потребностей. Каждому  $i$ -му изделию также соответствует набор значений параметров, характеризующих его эксплуатационные, технические, эргономические и другие свойства. Так как каждый набор значений параметров соответствует изделию с определенным номером  $i$ , далее будем считать, что тип изделия полностью определяется своим порядковым номером.

Необходимо из множества  $I$  допустимых типов изделий выбрать такое подмножество  $I^* \subset I$  (оптимальный ряд изделий), которое полностью обеспечит множество  $J$  видов потребностей с минимальными затратами.

Варианты математической формализации этой задачи зависят от многих факторов. В частности, их вид определяется структурой областей применения изделий, т. е. свойствами потребностей (обладают ли они свойством связности или нет), выбором критерия оптимизации ряда, учетом дополнительных ограничений, неопределенностью спроса и т. п. [1].

В данной работе мы рассмотрим формализацию указанной задачи в следующем виде. Требуется определить

$$C = \min \left( \sum_{i=1}^M c_i^0 y_i + \sum_{i=1}^M \sum_{j=1}^N c_{ij} x_{ij} \right) \quad (1)$$

при ограничениях

$$\sum_{i=1}^M \alpha_{ij} x_{ij} = 1, \quad j = 1, 2, \dots, N; \quad (2)$$

$$x_{ij} \leq y_i, \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N; \quad (3)$$

$$\begin{aligned} x_{ij} \in \{0, 1\}, \quad y_i \in \{0, 1\}, \\ i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N; \end{aligned} \quad (4)$$

$$\sum_{i=1}^M y_i \leq M_0, \quad M_0 \leq M, \quad (5)$$

где  $c_i^0$  — постоянные затраты на разработку и применение изделий  $i$ -го типа (начальные затраты);

$c_{ij} = \alpha_{ij} c_i b_j$  — затраты, связанные с обеспечением потребностей  $j$ -го вида изделиями  $i$ -го типа;

$\alpha_{ij}$  — число изделий  $i$ -го типа, необходимых для обеспечения единицы потребностей  $j$ -го вида;

$c_{ij}$  — затраты на производство и эксплуатацию одного изделия  $i$ -го типа;

$b_{ij}$  — величина потребностей  $j$ -го вида;

$M_0$  — количество типов изделий, которые могут быть включены в оптимальный ряд  $I^*$ .

Взаимосвязь элементов множеств  $I$  и  $J$  задается с помощью матрицы применений  $\|\alpha_{ij}\|_{M \times N}$ , в которой  $\alpha_{ij} = 1$ , если изделие  $i$ -го типа может обеспечивать  $j$ -й вид потребности, и  $\alpha_{ij} = 0$  в противном случае. Причем полагаем, что  $c_{ij} = \infty$ , если  $\alpha_{ij} = 0$ .

Управляющие переменные принимают значения:

$$y_i = \begin{cases} 1, & \text{если } i\text{-й тип изделия} \\ & \text{включают в выбираемый} \\ & \text{ряд изделий;} \\ 0 & \text{в противном случае;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й тип изделий} \\ & \text{назначен для удовлетворения} \\ & \text{потребности } j\text{-го вида;} \\ 0 & \text{в противном случае.} \end{cases}$$

Ограничения (2) означают, что все потребности множества  $J$  должны быть обеспечены. Из выражения (3) следует, что для обеспечения потребностей  $j$ -го вида можно назначать изделия только тех типов, для которых  $y_j = 1$ .

Соотношения (1)–(5) формализуют рассматриваемую задачу в виде модели дискретной многомерной задачи стандартизации при детерминированном спросе [2].

### Алгоритм решения задачи с использованием схемы метода ветвей и границ и теории двойственности

Рассмотрим алгоритм решения задачи (1)–(5), представляющий собой процесс направленного перебора с возвращениями. В основу этого процесса положена принципиальная схема метода ветвей и границ. При этом в предлагаемом алгоритме построение дерева возможных вариантов осуществляется в соответствии с односторонней схемой ветвления [2]. Для оценки нижней грани-

цы решения в процессе ветвления используется задача линейного программирования, двойственная по отношению к задаче (1)–(5), в которой условие целочисленности (4) заменяется условием

$$0 \leq y_i \leq 1, \quad i = 1, 2, \dots, M. \quad (6)$$

Проверка допустимого решения задачи (1)–(5) на оптимальность осуществляется решением оценочной двойственной задачи.

Для описания алгоритма введем следующие обозначения:

$I_0 = \{i | y_i = 0\}$  — множество индексов управляющих переменных  $y_i$ , вошедших в ветвь дерева вариантов со значением, равным нулю (такие переменные будем обозначать через  $y_i$ );

$I_1 = \{i | y_i = 1\}$  — множество индексов управляющих переменных  $y_i$ , вошедших в ветвь дерева вариантов со значением, равным единице;

$I_2 = I_1 \cup I_0$  — множество индексов управляющих переменных  $y_i$ , вошедших в ветвь дерева вариантов;

$I_3 = I/I_2$  — множество индексов управляющих переменных  $y_i$ , из которых выбирается индекс переменной для включения в множество  $I_2$ ;

$I_d$  ( $I_d = I_1$  при условии  $|I_2| = M$  или  $I_3 = \emptyset$ ) — множество индексов управляющих переменных  $y_i$ , соответствующих допустимому решению задачи (1)–(5);

$i^+$  — индекс управляющей переменной  $y_i$ , включение которого в множество  $I_1$  предпочтительнее по сравнению с другими индексами из множества  $I_3$ ;

$i^-$  — индекс управляющей переменной  $y_i$ , который целесообразно включить в множество  $I_0$ .

Значения  $c_{ij} = \infty$  заменяем величинами, равными  $\min_{i \in I} (c_i^0 + c_{ij})$ .

Двойственную задачу линейного программирования по отношению к релаксированной задаче (1)–(3), (6) можно записать следующим образом:

$$Z = \max \sum_{j \in J} V_j \quad (7)$$

при ограничениях

$$\sum_{j \in J} W_{ij} \leq c_i^0, \quad i = 1, 2, \dots, M; \quad (8)$$

$$V_j - W_{ij} \leq c_{ij}, \quad j = 1, 2, \dots, N, \quad (9)$$

$$i = 1, 2, \dots, M;$$

$$W_{ij} \geq 0, \quad j = 1, 2, \dots, N, \quad (10)$$

$$i = 1, 2, \dots, M,$$

где  $V_j, W_{ij}$  — переменные двойственной задачи.

Теоретически задачу (7)–(10), введя дополнительные переменные  $\gamma_{ij}, \delta_i$ , ( $i = 1, M, j = 1, N$ ), можно представить в виде основной задачи линейного программирования, исходное базисное решение которой имеет вид

$$Z = \sum_{j=1}^N V_j; \gamma_{ij} = c_{ij} + W_{ij} - V_j,$$

$$i = 1, 2, \dots, M, j = 1, 2, \dots, N;$$

$$\delta_i = c_i^0 - \sum_{j=1}^N W_{ij}, i = 1, 2, \dots, M.$$

Полученную таким образом задачу можно решить симплексным методом.

Однако практика показала, что даже небольшие задачи оптимизации многомерных параметрических рядов требуют решения задач линейного программирования большой размерности. Их опорные планы сильно вырождены, а установление оптимальности в некоторых случаях связано с большим числом симплексных итераций. Действительно, количество переменных  $k_1$  и число уравнений ограничений  $k_2$  задачи линейного программирования связано с размерностью исходной задачи соотношениями  $k_1 = M + 2MN + N$ ,  $k_2 = M(N + 1)$ . Для решения задачи линейного программирования такой размерности требуется обычно от  $M(N + 1)$  до  $3M(N + 1)$  итераций. Учитывая, что оценка нижней границы производится многократно, такой способ решения является нерациональным. Кроме того, нецелесообразно искать и точное решение задачи (7)–(10), так как оно позволяет получить точное решение не исходной целочисленной задачи (1)–(5), а лишь непрерывной релаксированной задачи (1)–(3), (6). Поэтому дальнейшее развитие рассматриваемого направления идет по пути построения приближенных методов решения двойственной задачи [3] и упрощения тем самым способов оценки нижней границы исходной целочисленной задачи. Алгоритмы, использующие для оценки двойственную задачу, различаются способами определения приближенного решения задачи (7)–(10). Наиболее простым является способ расчета, основанный на следующих соотношениях:

$$W_{ij} = \frac{c_i^0}{J_i}; V_j = \min_{i \in I_j} (c_{ij} + W_{ij}); Z = \sum_{j=1}^N V_j,$$

где  $J_i = \{j | \alpha_{ij} = I\}$ ;  $I_j = \{i | \alpha_{ij} = I\}$ .

При этом достаточно простая структура задачи (7)–(10) позволяет получить приближенное решение, совпадающее с оптимальным или достаточно близкое к нему. Вычислительные эксперименты на ЭВМ показали, что особенностью задачи (1)–(5) является довольно частое совпадение оптимального целочисленного решения с непрерывным. Поэтому обычно приближенно решают двойственную задачу, а от нее переходят к соответствующему решению непрерывной задачи (1)–(3), (6). При совпадении целочисленного

и непрерывного решений полученный результат является оптимальным, в противном случае переходят к обычному алгоритму ветвей и границ. Однако переход от приближенного решения двойственной задачи к соответствующему приближенному решению задачи (1)–(5) является достаточно сложным и не всегда однозначным. Значительно более простым является способ, когда вначале определяется достаточно точное решение задачи (1)–(5), а по этому решению определяется решение двойственной задачи. Совпадение решений свидетельствует об оптимальности целочисленного решения. В дальнейшем этот способ мы будем называть направленным, так как он использует решение целочисленной задачи в оцениваемой вершине.

Рассмотрим направленный способ решения двойственной задачи (7)–(10) более подробно.

Упростим структуру задачи (7)–(10), для этого исключим переменные  $W_{ij}$ , входящие лишь в ограничения. Действительно, при любом допустимом выборе переменных  $V_j$  переменные  $W_{ij}$  можно положить равными максимальной из величин 0 и  $V_j - c_{ij}$ . С учетом этого двойственную задачу (7)–(10) запишем в виде

$$Z = \max \sum_{j=1}^N V_j \quad (11)$$

при ограничениях

$$\sum_{j=1}^N \max(V_j - c_{ij}; 0) \leq c_i^0,$$

$$i = 1, 2, \dots, M. \quad (12)$$

Сущность направленного решения двойственной задачи (11), (12) заключается в следующем. Учитывая ее более простую структуру и возможность использовать приближенное решение для оценки нижней границы на различных этапах ветвления, предлагаем следующий способ формирования значений переменных  $V_j$ , доставляющих целевой функции (11) по возможности большее значение. Процесс начинается с допустимого решения  $V_j = \min_{i \in I_1 \in I_3} c_{ij}$  задачи (11), (12) и состоит

в многократном последовательном просмотре  $j$ -х столбцов ( $j = 1, 2, \dots, N$ ) матрицы  $\|c_{ij}\|$  в целях проверки возможности увеличения  $V_j$  до ближайшей большей величины  $c_{ij}$ . При этом в каждом цикле просмотра проверяется возможность увеличения  $V_j$  для тех столбцов, в которых находится наименьшее количество элементов  $c_{ij}$ , удовлетворяющих условию  $c_{ij} \leq V_j$ . Увеличение значений переменных  $V_j$  происходит с учетом (12) и ограничено величинами

$$V_j \leq \min_{i \in I_1} [\min_{i \in I} c_{ij}; \min_{i \in I} (c_{ij} + c_i^0)]. \quad (13)$$

Оценка, получаемая с помощью данного способа, определяется элементами множеств  $I_1, I_3$  и, когда дальнейшее увеличение переменных  $V$  становится невозможным, вычисляется по формуле

$$Z(I_1) = z_x = \sum_{j=1}^N V_j + \sum_{i \in I_1} c_i^0. \quad (14)$$

В ходе экспериментов со случайными матрицами  $\|c_{ij}\|$  выявлено довольно частое (до 90 %) совпадение оптимального целочисленного решения задачи (1)–(5) с решением, которое определялось с помощью двойственной задачи (11), (12). Учет этой особенности в сочетании с достаточно простым способом нахождения решения двойственной задачи позволил значительно сократить время поиска оптимального решения задачи (1)–(5). Выигрыш во времени достигается за счет доказательства оптимальности одного из допустимых решений задачи (1)–(5), полученных на первых шагах работы алгоритма.

Эффективность применения данного способа и точность полученного с его помощью результата в значительной степени определяются точностью используемого (одновременно проверяемого на оптимальность) целочисленного решения. Если для оценки нижней границы целевой функции путем направленного решения двойственной задачи (11), (12) используются ограничения (13), то при проверке допустимого решения на оптимальность вместо (13) требуется выполнение неравенств

$$c_{rj} \leq V_j \leq \min [c_{ij}; \min_{i \in I} (c_{ij} + c_i^0)], \quad j = 1, 2, \dots, N, \quad (15)$$

где  $c_{ri} = \min_{i \in I_d} c_{ij}$ ;  $c_{ij} = \min_{i \in I, i \neq 1} c_{ij}$ .

Процесс проверки допустимого решения задачи (1)–(5) на оптимальность начинается с допустимого решения  $V_j = \min_{i \in I} c_{ij}$  задачи (11), (12).

Необходимость неравенств (15) следует из соотношений

$$y_i^* \left[ c_i^0 - \sum_{j=1}^N \max(V_j^* - c_{ij}; 0) \right] = 0; \\ (y_i^* - x_{ij}^*) \max(V_j^* - c_{ij}; 0) = 0,$$

определяющих оптимальность полученных решений для непрерывной и целочисленной задач. Знак «\*» указывает на оптимальность решений соответствующих задач. При выполнении ограничения (15) совпадение величины

$$Z(I_d) = \sum_{j=1}^N V_j, \quad (16)$$

полученной в результате решения двойственной задачи (11), (12), со значением целевой функции

задачи (1)–(5), соответствующей проверяемому целочисленному решению

$$C_d = \sum_{i \in I_d} c_i^0 + \sum_{j=1}^N \min_{i \in I_d} c_{ij}, \quad I_d = I, \quad (17)$$

указывает на оптимальность найденного целочисленного решения, т. е.

$$I^* = I_d, \quad c^* = c_d.$$

В случае несовпадения значений целевых функций поиск оптимального решения задачи (1)–(5) продолжается методом ветвей и границ.

Поиск первого допустимого решения и дальнейшее ветвление осуществляется с помощью показателей вида

$$\rho_i = \sum_{j=1}^N \max(V_j - c_{ij}; 0) - c_i^0, \quad i \in I_3, \quad (18)$$

где  $V_j = \min \left[ \min_{i \in I_1} c_{ij}; \min_{i \in I} (c_i^0 + c_{ij}) \right]$ .

Таким образом, общая схема алгоритма решения задачи (1)–(5) включает следующие основные этапы:

- 1) определение допустимого решения  $I_d, C_d$  исходной задачи;
- 2) проверка допустимого решения на оптимальность путем решения двойственной задачи;
- 3) уточнение, в случае необходимости, допустимого решения методом ветвей и границ и проверка его на оптимальность.

Вычислительный процесс заканчивается при выполнении одного из условий:

- доказана оптимальность допустимого решения с помощью решения двойственной задачи;
- просмотрены все перспективные ветви дерева вариантов.

Перед началом решения задачи полагаем  $I_0 = I_1 = I_2 = 0, I_3 = \{1, 2, \dots, M\}$ .

На каждом шаге вычислительного процесса определяются показатели  $\rho_i, i \in I_3$ , характеризующие целесообразность включения переменных  $y_i, i \in I_3$  в формируемое допустимое решение задачи (1)–(5). Если среди показателей (18) имеются  $\rho_i < 0$ , то принимаем соответствующие им переменные  $y_i$  равными нулю, так как включение их в допустимое решение не приведет к его улучшению. Индексы  $\{i | \rho_i < 0\}$  выводим из множества  $I_3$  и включаем в множество  $I_0$ . Кроме того, из множества  $I_3$  исключается и в множество  $I_1$  вводится индекс  $i^+$ , для которого

$$\rho_{i^+} = \max_{\{i | \rho_i \geq 0\}} \rho_i, \quad (19)$$

так как переменная  $y_{i^+}$  может привести к улучшению решения. Последним шагом алгоритма, заканчивающим формирование допустимого ре-

шения, является шаг, на котором  $I_3 = 0$ . Значение целевой функции, соответствующее найденному допустимому целочисленному решению, определяется согласно (17). Если проверка с помощью решения двойственной задачи не подтверждает оптимальности допустимого решения, то его уточняем методом ветвей и границ. Схема ветвления односторонняя, использующая правило «последний вошел — первый выходит», т. е. для ветвления выбирается висячая вершина, образованная последней. В основу дерева вариантов принимается ветвь, полученная на первом этапе, дальнейшее ветвление производится с помощью показателей  $r_i$ , как и при поиске первого допустимого решения.

Отсечение бесперспективных ветвей осуществляется сравнением оценки (14), полученной в результате решения двойственной задачи (11)—(12) с учетом ограничения (13), со значением целевой функции (17), соответствующей наилучшему из уже найденных допустимых решений. Поэтому здесь важно с самого начала получить значение целевой функции задачи (1)—(5), близким к оптимальному. Следовательно, способ получения допустимого решения должен быть достаточно эффективным.

Одной из основных сложностей, связанных с решением задач оптимизации многомерных параметрических рядов, является их большая размерность. В некоторых случаях размерность задачи можно уменьшить на этапе предварительного анализа исходных данных. Сущность одного из возможных способов заключается в определении с помощью специальных показателей тех изделий, которые обязательно следует включить в оптимальный ряд. Поиск таких изделий осуществляется без решения исходной оптимизационной задачи с помощью показателей

$$\Delta_i = \sum_{\substack{l=1 \\ l \neq i}}^N \min [\max(c_{lj} - c_{ij}; 0)] - c_i^0, \quad i \in I, \quad (20)$$

характеризующих увеличение затрат на обеспечение заданных потребностей в изделиях, если принять  $y_i = 0$ . Поэтому при условии  $\Delta_i \geq 0$  необходимо включить  $i$ -й тип изделия в  $I^*$ , приняв  $y_i = 1$ .

В целом предложенный подход к математической формализации задачи формирования параметрических рядов и алгоритм ее решения позволяют обеспечить рациональную интеграцию инновационных проектов в предметную область их использования и могут найти широкое применение в инновационном менеджменте.

## Литература

1. Береснев В. Л., Гимади Э. Х., Дементьев В. Г. Экстремальные задачи стандартизации. — Новосибирск: Наука, 1978. — 333 с.
2. Анисимов В. Г., Анисимов Е. Г. Метод решения одного класса задач целочисленного программирования // Журнал вычислительной математики и математической физики. 1989. Т. 29. № 10. С. 1586–1590.
3. Алексеев О. Г., Анисимов В. Г., Анисимов Е. Г. Применение двойственности для повышения эффективности метода ветвей и границ при решении задачи о ранце // Журнал вычислительной математики и математической физики. 1985. Т. 25. № 11. С. 1666–1673.