УДК 330.1, 65.0

В. А. Мещеряков

ФОРМАЛИЗАЦИЯ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ ОТВЕТСТВЕННОСТИ И ПОЛНОМОЧИЙ МЕЖДУ ПОДРАЗДЕЛЕНИЯМИ И СОТРУДНИКАМИ УНИВЕРСИТЕТА

Осуществлена формализации процесса распределения ответственности и полномочий между подразделениями и сотрудниками университета в рамках системы менеджмента качества (СМК), на основе которой решается задача отыскания оптимального управления с привлечением математического аппарата метода динамического программирования.

Введение

При управлении системой менеджмента качества (СМК) университета для результативного принятия управленческих решений необходимо формализовать процесс распределения ответственности и полномочий между подразделениями и сотрудниками университета. При этом необходимо, во-первых, чтобы все требования, предъявляемые к СМК, были выполнены оптимальным образом, т.е. с привлечением минимальных ресурсов (человеческих, материальных, финансовых, информационных и др.), и, в то же время, результативно; во-вторых, сформулировать критерии оценки оптимальности управления и рассмотреть различные подходы к их формулировке.

Формализация задачи управления в СМК

Представим требования к СМК (например, по ГОСТ Р ИСО 9001–2001) в виде

$$x_i(r_{i,1}, r_{i,2}, ..., r_{i,l}),\ i = 1, 2, ..., m, \tag{1}$$

где координаты $r_{i,1}, r_{i,2}, ..., r_{i,l}$ представляют собой требуемые ресурсы для выполнения требований (1).

Для каждого из заданных требований на каждом из уровней подчиненности находим оптимального исполнителя из ряда

$$y_j^k(R_{j,1}^k, R_{j,2}^k, ..., R_{j,l}^k),\ j = 1, 2, ..., n, \tag{2}$$

где координаты $R_{j,1}^k, R_{j,2}^k, ..., R_{j,l}^k$ определяют ресурсы, которыми располагает j – исполнитель подчиненности уровня k $(k = 1, 2, ..., K)$.

В государственном образовательном учреждении (университете) под уровнями подчиненности понимаются:

– общеуниверситетский уровень;

– уровень института/факультета/управления;

– уровень кафедры/структурного подразделения.

Данная задача является задачей отыскания оптимального управления – обратной задачей динамики дискретного типа, которая рассмотрена в [1]. Она носит дискретный характер, является обратной стационарной задачей динамики, т.е. предполагается, что ее решение не зависит от времени. Поставленную задачу будем рассматривать как линейную задачу управления.

Обобщенная математическая модель процедуры оптимального управления СМК будет описываться уравнением

$$x = f(x, u). \tag{3}$$

В качестве функции управления рассматривается ее дискретный аналог – трехмерная матрица соответствий:

$$U = \{u_{i,j}^k\}, \tag{4}$$

где ее элементы есть экспертные оценки предпочтений: для каждого исполнителя $(y_j^k - k)$-го уровня и каждого требования $x_i(r_{i,1}, r_{i,2}, ..., r_{i,l})$ ставится в соответствие элемент данной матрицы.

Для решения данной задачи введем ограничения.

Ограничение 1 – по искомым ресурсам:

$$\begin{cases} R_{j,s}^k \geq 0, \\ r_{i,s} \geq 0, \\ R_{j,s}^k - r_{i,s} \geq 0, \\ k = 1, 2, ..., K, i = 1, 2, ..., m, \\ j = 1, 2, ..., n, s = 1, 2, ..., l. \end{cases} \tag{5}$$

Ограничение 2 – по исполнителям:

а) на каждом уровне должен быть единственный исполнитель:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} h_{i,j}^k u_{i,j}^k = 1, \tag{6}$$

где $H = \{h_{i,j}^k\} = \{0, 1\}$ – множество значений трехмерной матрицы – либо ноль, либо единица;

б) для каждого требования должно быть K исполнителей:

$$\sum_{j=1}^{n} \sum_{k=1}^{K} h_{i,j}^k u_{i,j}^k = K. \tag{7}$$

Введем критерии оптимальности.

Критерий 1 – оптимальность по качеству используемых ресурсов:

$$\begin{cases} R_{j,s}^k \geq 0, \\ r_{i,s} \geq 0, \\ R_{j,s}^k - r_{i,s} \to \min, \\ k = 1, 2, ..., K, i = 1, 2, ..., m, \\ j = 1, 2, ..., n, s = 1, 2, ..., l. \end{cases} \tag{8}$$

Данное условие требует при одном и том же управлении как можно большей прилежности данного исполнителя на исполнение своих обязанностей.

Критерий 2 – оптимальность по экономии ресурсов:

$$\begin{cases} R_{j,s}^k \ge 0, \\ r_{i,s} \ge 0, \\ R_{j,s}^k - r_{i,s} \to \max, \\ k = 1, 2, ..., K, \; i = 1, 2, ..., m, \\ j = 1, 2, ..., n, \; s = 1, 2, ..., l. \end{cases} \tag{9}$$

Данное условие требует при одном и том же управлении как можно больше сэкономить ресурсы, например материальные.

Критерий 3 – оптимальность по чувствительности требования x_i к ресурсу $r_{i,p}$:

$$\frac{\partial x_i}{\partial r_{i,p}} \to \max \text{ или } \frac{\partial x_i}{\partial r_{i,p}} \to \min . \tag{10}$$

Критерий 4 – для заданного требования и найденных исполнителей проверяется условие

$$\frac{\sum_{k=1}^{K} x_i \cdot y_k^k}{\sqrt{\sum x_i^2 \sqrt{\sum y_k^k}}} = \frac{\sum_{k=1}^{K} (r_{i,1}R_{k,1}^k + r_{i,2}R_{k,2}^k + ... + r_{i,s}R_{k,s}^k)}{\sqrt{\sum x_i^2 \sqrt{\sum y_k^k}}} \approx K. \tag{11}$$

Данное условие означает, что вектора $x_i(r_{i,1}, r_{i,2}, ..., r_{i,l})$, $y_j^k(R_{j,1}^k, R_{j,2}^k, ..., R_{j,l}^k)$ «почти» коллинеарные, и, следовательно, есть большая вероятность, что интересы в рассматриваемом требовании совпадают с возможностями данного исполнителя. Этот критерий позволяет также пропорционально расходовать ресурсы данного исполнителя по требованию, что с большой вероятностью эффективно скажется на его работе.

При этом данный критерий используется при формировании матрицы (4):

$$u_{i,j}^k = x_i y_j^k = \left(r_{i,1}R_{j,1}^k + r_{i,2}R_{j,2}^k + ... + r_{i,s}R_{j,s}^k \right).$$

Отметим, что скалярный критерий оптимальности наиболее эффективен при многошаговом процессе решения, например при применении динамического программирования. Он позволяет найти опорный план и сделать шаг к последующему решению от предыдущего, где на каждом шаге проверяют один из критериев оптимальности или их совокупности [2].

Определение оптимального управленческого решения распределения полномочий и ответственности между исполнителями в СМК

Используем метод динамического программирования, который излагается согласно [2].

Будем считать, что состояние рассматриваемой системы S на k-м шаге $\left(k=\overline{1,n}\right)$ определяется совокупностью чисел $X^{(k)}=\left(x_1^{(k)},x_1^{(k)},...,x_n^{(k)}\right)$, которые получены в результате реализации матрицы управления

$$U^k=\begin{pmatrix} u_{1,1}^k & u_{1,2}^k & ... & u_{1,n}^k \\ u_{2,1}^k & u_{2,2}^k & ... & u_{2,n}^k \\ . & . & . & . \\ u_{m,1}^k & u_{m,2}^k & ... & u_{m,n}^k \end{pmatrix} \qquad (12)$$

обеспечивающей переход системы S из состояния $X^{(k-1)}$ в состояние $X^{(k)}$. При этом будем предполагать, что состояние $X^{(k)}$, в которое перешла система S, зависит от данного состояния $X^{(k-1)}$ и выбранного управления u_k и не зависит от того, каким образом система S пришла в состояние $X^{(k-1)}$.

Далее будем считать, что если в результате реализации k-го шага обеспечена требуемая результативность процесса, также зависящая от исходного состояния системы $X^{(k-1)}$, выбранного управления u_k и равная $W_k\left(X^{(k-1)},u_k\right)$, то общая результативность за n шагов составляет:

$$F=\sum_{k=1}^{n}W_k\left(X^{(k-1)},u_k\right). \qquad (13)$$

Таким образом, мы сформулировали два условия, которым должна удовлетворять рассматриваемая задача динамического программирования. Первое условие обычно называют условием отсутствия последействия, а второе – условием аддитивности целевой функции задачи.

Выполнение для задачи динамического программирования первого условия позволяет сформулировать для нее принцип оптимальности Беллмана. Прежде чем сделать это, дадим определение оптимальной стратегии управления. Под такой стратегией будем понимать совокупность управлений $U^*=\left(u_1^*,u_2^*,...,u_n^*\right)$, в результате реализации которых система S за n шагов переходит из начального состояния $X^{(0)}$ в конечное $X^{(n)}$ и при этом функция (13) принимает наибольшее значение.

Каково бы ни было состояние системы перед очередным шагом, надо выбрать управление на этом шаге так, чтобы результативность на данном шаге плюс оптимальная результативность на всех последующих шагах была максимальной.

Отсюда следует, что оптимальную стратегию управления можно получить, если сначала найти оптимальную стратегию управления на m-м шаге, затем на двух последних шагах, затем на трех последних шагах и т.д., вплоть до первого шага. Таким образом, решение рассматриваемой задачи динамического программирования целесообразно начинать с определения оптимального решения на последнем, m-м шаге. Для того чтобы найти это решение, очевидно, нужно сделать различные предположения о том, как мог окончиться предпоследний шаг, и с учетом этого выбрать управление u_n^0, обеспечивающее максимальное значение функции $W_n\left(X^{(n-1)}, u_n\right)$. Такое управление u_n^0, выбранное при определенных предположениях о том, как окончился предыдущий шаг, называется условно оптимальным управлением. Следовательно, требуется находить на каждом шаге условно оптимальное управление для любого из возможных исходов предшествующего шага.

Чтобы это можно было осуществить практически, необходимо математически сформулировать принцип оптимальности. Для этого введем некоторые дополнительные обозначения. Обозначим через $F_n\left(X^{(0)}\right)$ максимальную результативность, получаемую за n шагов при переходе системы S из начального состояния X^0 в конечное состояние $X^{(n)}$ при реализации оптимальной стратегии управления $U^* = \left(u_1^*, u_2^*, ..., u_n^*\right)$, а через $F_{n-k}\left(X^{(k)}\right)$ – максимальную результативность, получаемую при переходе из любого состояния $X^{(k)}$ в конечное состояние $X^{(n)}$ при оптимальной стратегии управления на оставшихся $n-k$ шагах. Тогда

$$F_n\left(X^{(0)}\right) = \max_{u_{k+1}}\left[W_1\left(X^{(0)}, u_1\right) + ... + W_n\left(X^{(n-1)}, u_n\right)\right]; \qquad (14)$$

$$F_{n-k}\left(X^{(k)}\right) = \max_{u_{k+1}}\left[W_{k+1}\left(X^{(k)}, u_{k+1}\right) + ... + F_{n-k-1}\left(X^{(k+1)}\right)\right], \qquad (15)$$

$$k = \overline{0, n-1}.$$

Последнее выражение представляет собой математическую запись принципа оптимальности и носит название основного функционального уравнения Беллмана или рекуррентного соотношения. Используя данное уравнение, находим решение рассматриваемой задачи динамического программирования. Остановимся на этом более подробно.

Полагая $k = n - 1$ в рекуррентном соотношении (15), получаем следующее функциональное уравнение:

$$F_1\left(X^{(n-1)}\right) = \max_{u_n}\left[W_n\left(X^{(n-1)}, u_n\right) + F_0\left(X^{(n)}\right)\right]. \qquad (16)$$

В этом уравнении $F_0\left(X^{(n)}\right)$ будем считать известным. Используя теперь уравнение (16) и рассматривая всевозможные допустимые состояния системы S на (n–1)-м шаге $X_1^{(n-1)}$, $X_2^{(n-1)}$, …, $X_m^{(n-1)}$, …, находим условные оптимальные решения:

$$u_n^0\left(x_1^{(n-1)}\right), u_n^0\left(x_2^{(n-1)}\right), ..., u_n^0\left(x_m^{(n-1)}\right), ...$$

и соответствующие значения функции (16):

$$F_1^0\left(x_1^{(n-1)}\right), F_1^0\left(x_2^{(n-1)}\right), ..., F_1^0\left(x_m^{(n-1)}\right),...$$

Таким образом, на n-м шаге находим условно оптимальное управление при любом допустимом состоянии системы S после (n–1)-го шага. Иными словами, в каком бы состоянии система ни оказалась после (n–1)-го шага, нам уже известно, какое следует принять решение на n-м шаге. Известно также и соответствующее значение функции (16).

Переходим теперь к рассмотрению функционального уравнения при $k = n–2$:

$$F_2\left(X^{(n-2)}\right) = \max_{u_{n-1}}\left[W_{n-1}\left(X^{(n-2)}, u_{n-1}\right) + F_1\left(X^{(n-1)}\right)\right]. \quad (17)$$

Для того чтобы найти значения F_2 для всех допустимых значений $X^{(n-2)}$, очевидно, необходимо знать $W_{n-1}\left(X^{(n-2)}, u_{n-1}\right)$ и $F_1\left(X^{(n-1)}\right)$. Что касается значений $F_1\left(X^{(n-1)}\right)$, то мы их уже определили. Поэтому нужно произвести вычисления для $W_{n-1}\left(X^{(n-2)}, u_{n-1}\right)$ при некотором наборе допустимых значений $X^{(n-2)}$ и соответствующих управлениях u_{n-1}. Эти вычисления позволят определить условно оптимальное управление u_{n-1}^0 для каждого $X^{(n-2)}$. Каждое из таких управлений совместно с уже выбранным управлением на последнем шаге обеспечивает максимальную результативность на двух последних шагах.

Последовательно осуществляя описанный выше итерационный процесс, дойдем, наконец, до первого шага. На этом шаге нам известно, в каком состоянии может находиться система. Поэтому уже не требуется делать предположений о допустимых состояниях системы, а остается лишь только выбрать управление, которое является наилучшим с учетом условно оптимальных управлений, уже принятых на всех последующих шагах.

Таким образом, в результате последовательного прохождения всех этапов от конца к началу определяем максимальное значение результативности за n шагов и для каждого из них находим условно оптимальное управление.

Чтобы найти оптимальную стратегию управления, т.е. определить искомое решение задачи, нужно теперь пройти всю последовательность шагов, только на этот раз от начала к концу. А именно: на первом шаге в качестве оптимального управления u_1^* возьмем найденное условно оптимальное управление u_1^0. На втором шаге найдем состояние X_1^*, в которое переводит систему управление u_1^*. Это состояние определяет найденное условно оптимальное управление u_2^0, которое теперь будем считать оптимальным. Зная u_2^*, находим X_2^*, а значит, определяем u_3^* и т.д. В результате этого находим решение задачи, т.е. максимально возможную результативность и оптимальную стратегию управления U^*, включающую оптимальные управления на отдельных шагах: $U^*\left(u_1^*, u_2^*, ..., u_n^*\right)$.

По найденной матрице (12) строится матрица H, которая, в конечном счете, и будет выражать соответствие между данным требованием и найденными исполнителями. Условие, по которому строится матрица H-соответствий, может быть одним из следующих:

– для каждого уровня k матрицы U находится максимальный ее элемент:

$$h_{i,j}^{k} = \begin{cases} 1, \max\limits_{i,j} u_{i,j}^{k}, \\ 0, \quad \text{в противном случае;} \end{cases}$$

– определяется по скалярному критерию (11):

$$h_{i,j}^{k} = \begin{cases} 1, x_i \cdot y_j^k = (r_{i,1}R_{j,1}^{k} + r_{i,2}R_{j,2}^{k} + \ldots + r_{i,s}R_{j,s}^{k}) \approx 1, \\ 0, \quad \text{в противном случае.} \end{cases}$$

Матрица H^k для k-го уровня может быть, например, такой:

$$H^{k} = \begin{pmatrix} 0 & 1 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ . & . & . & . \\ 0 & 0 & \ldots & 1 \end{pmatrix}.$$

Например, на k-м уровне за исполнение первого требования отвечает второй исполнитель и т.д.

Заключение

На основе введенных: ограничений по искомым ресурсам и исполнителям; критериев оптимальности по качеству, экономии используемых ресурсов, чувствительности требования к ресурсу, скалярному критерию, – определена матрица H-соответствий между конкретным требованием и найденными исполнителями.

Список литературы

1. **Акулич, И. Л.** Математическое программирование в примерах и задачах : учебное пособие / И. Л. Акулич. – 2-е изд., испр. и доп. – М. : Высш. шк., 1993. – 336 с.
2. **Беллман, Р.** Динамическое программирование / Р. Беллман. – М. : Изд-во иностр. лит., 1960. – 400 с.