

РАЗРАБОТКА АЛГОРИТМА ОПТИМАЛЬНОГО УПРАВЛЕНИЯ ПО УСТРАНЕНИЮ НЕПРОИЗВОДИТЕЛЬНЫХ ПОТЕРЬ

© 2010 Е.А. Сидорова*

Ключевые слова: оптимальное управление, потеря, затраты, алгоритм, объем выпуска, матрица, переменные.

Приводится классификация потерь на основании теоретических положений системы Lean Production. В соответствии с классификацией разработан алгоритм оптимального устранения потерь в производственной цепочке. Описаны матрицы состояний системы, требований клиента и управляющая матрица.

В условиях кризиса актуальной является проблема устранения непроизводственных затрат (потерь), так как это снижает себестоимость продукции и увеличивает рентабельность производства. Целью работы является построение алгоритма оптимизации потерь. В основе алгоритма лежит принцип оптимального управления Беллмана. В развитии алгоритма сформулирован ряд условий и подусловий проверки информации о виде потерь. Исходя из установленного вида потери, принимается стандартное управление. Данный алгоритм записан в виде блок-схем с использованием языка описаний и спецификаций (Specification and Description Language, SDL) на основе спецификаций ССИТТ. Описаны как внутренние, так и внешние переменные в составе алгоритма.

Разработанный нами алгоритм, основан на выделении семи основных видов потерь согласно классификации системы Lean Production¹, а именно:

1. Перепроизводство.
2. Лишние запасы.
3. Потери из-за выпуска дефектных изделий.
4. Ожидания и простои.
5. Ненужная транспортировка.
6. Лишние этапы обработки (работа, не добавляющая ценности).
7. Неэффективность производственных операций (потери при выполнении операций).

Вместе с тем необходимо отметить, что под потерями в общем виде понимаются все действия, производимые над изделием, не добавляющие ценности относительно требо-

ваний клиента². Исходя из данного уточнения, можно предположить, что в случае устранения всех потерь, производимая продукция полностью будет соответствовать предпочтениям клиента.

Алгоритм оптимизации потерь заключается во взаимосвязи классификации потерь и состояния технологического процесса, представляющий собой процесс преобразования технологической цепочки. Процесс преобразования начинается с последней операции³ технологического процесса: мы выявляем потери, которые не позволяют удовлетворить требованиям клиента, и в зависимости от вида предлагаем инструмент для их устранения⁴. Аналогично, преобразуем остальные операции, передвигаясь в обратном направлении от последней до первой операции технологической цепочки, что, позволяет нам сформировать требования к собственным поставщикам⁵. Нужно также отметить, что на одной операции может возникать несколько видов потерь.

Механизм выявления потерь основывается на сравнении требований клиента и возможностей производителя. В то же время, требования потребителя и возможности производителя, при условии отсутствия потерь, должны совпадать, это следует из определения понятия потери, приведенного выше.

Исходное состояние системы описывается следующими показателями:

а) величина фактического объема выпуска (q);

б) соответствующие объему выпуска фактические затраты (c).

* Сидорова Екатерина Андреевна, аспирант Волжского университета им. Татищева, г. Тольятти.
E-mail: EA.Sidorova@mail.ru.

Состояние системы зависит от времени. Это объясняется влиянием случайных факторов, таких как выход из строя оборудования, выпуск бракованной продукции и т.п. Поэтому необходимо провести ряд фиксаций состояний системы для получения независимых данных, которое составит i . Тогда, состояние системы опишется в виде матриц состояний объема выпуска и затрат соответственно:

и $P_{i \times n}$, где n - количество операций технологического процесса.

Обозначим матрицу управления U , где n - количество операций технологического процесса. Условимся, что для каждого вида потерь, применим конкретный набор управлений. Обозначим каждый набор управлений по устранению конкретного вида потерь в виде констант: $u_{k1}, u_{k2}, \dots, u_{k7}$ соответственно⁶. На одном шаге может быть применено несколько управлений в зависимости от количества видов потерь, выявленных на данном шаге. Итоговое управление на этом шаге будет представлять собой сумму управлений по каждому виду потерь.

Требования клиента к технологической цепочке также могут быть охарактеризованы фактическим объемом выпуска ($q_{клиента}$) - объем поставки и соответствующих этому объему фактических затрат ($p_{клиента}$) - цена поставки. Причем данные требования сравнимы с последней, т.е. n -й операцией технологического процесса. Так как частота изменений требований клиента незначительна по сравнению с колебанием состояния технологического процесса, то данные принимаются в виде усредненных по времени значений. Поэтому их можно считать константами для данного технологического процесса, то есть результирующее состояние системы дискретно.

Перейдем к механизму выявления потерь⁷. Как уже упоминалось выше, анализ осуществляется с последней n -й операции технологического процесса и имеет следующий вид, состоящий из набора условий и подусловий:

1. $q_{1n} > q_{клиента}$, $q_{2n} > q_{клиента}$, ...
 $q_{in} > q_{клиента}$ - с этим вариантом связана потеря перепроизводства продукции, т.е. постав-

щик производит больше, чем это необходимо клиенту. Принимаемое управление должно быть направлено на устранение этого вида потерь, т.е. $u_n = u_{k1}$. Также возникают потери излишних запасов, $u_n = u_{k1} + u_{k2}$. Суммирование в данном случае возможно, так как выявляется более одного вида потерь на операции.

2. $\frac{\sum q_{in}}{i} < q_{клиента}$ в данном варианте

возможны три под условия:

2.1. $q_{1n} < q_{клиента}$, $q_{2n} < q_{клиента}$,

... $q_{in} < q_{клиента}$ - данное условие означает, что на рынке постоянный дефицит товара поставщика, т.е. поставщик не успевает производить товар. Вследствие чего потребитель вынужден простаивать и ожидать, а поскольку возникает дефицит, то сменить поставщика не представляется возможным. В этом случае принимаем управление равное $u_n = u_{k4}$, т.е. устраняем четвертый вид потерь, связанный с ожиданиями и простоями.

2.2. $q_{1n} < q_{клиента}$, $q_{2n} > q_{клиента}$,

... $q_{in} < q_{клиента}$ - данный вариант означает, что в условиях дефицита товара в отдельные промежутки времени осуществляется перепроизводство и потребитель успевает запастись товаром, т.е. потребитель вынужден делать запасы, а поскольку возникает дефицит, то сменить поставщика не представляется возможным. В этом случае принимаем управление на операции, равное $u_n = u_{k2}$, т.е. устраняем потери, связанные с лишними запасами.

2.3. $p_{1n} > p_{клиента}$, $p_{2n} > p_{клиента}$,

... $p_{in} > p_{клиента}$ - данный вариант означает, что в условиях дефицита товара на рынке, затраты на производство товара поставщиком, превышают ту стоимость, которую готов платить потребитель, т.е. поставщик допускает потери при выполнении операции или операции функционируют не эффективно. В этом случае принимаем управление на операции, равное $u_n = u_{k7}$, т.е. устраняем поте-

ри, связанные с неэффективностью производственных операций.

$$3. \frac{\sum_i q_{in}}{i} = q_{\text{клиента}}, \text{ т.е. размер партий и}$$

время поставки совпадает, но не совпадает стоимость партии:

$$3.1. \frac{\sum_i p_{in}}{\sum_i q_{in}} > \frac{p_{\text{клиента}}}{q_{\text{клиента}}} \text{ и } \Delta p_{in} \gg \Delta q_{in} -$$

условие показывает, что при незначительном росте объема производства продукции функция затрат резко возрастает, а переменные затраты на производство единицы продукции превышают покупательную способность клиента. Это может быть связано с большими затратами на транспортировку, когда стоимость доставки до потребителя делает все производство поставщика не рентабельным и поставщику не выгодно нести на себе такие расходы. Данный случай соответствует потерям при излишней транспортировке, поэтому принимаем управление на операции $u_n = u_{к5}$.

$$3.2. \frac{\sum_i p_{in}}{i} > 0, \text{ это значит, что произво-}$$

димый товар вообще не нужен потребителю ни за какую стоимость. То есть работа производителя не добавляет ценности товару в глазах потребителя. Данное условие описывает потери, связанные с работой, не добавляющей ценности. Принимаем $u_n = u_{к6}$.

3.3. $p_{1n} = p_{\text{клиента}}, p_{2n} > p_{\text{клиента}}, \dots p_{in} = p_{\text{клиента}}$, это значит, что в определенные промежутки времени затраты поставщика превышают стоимость, которую готов заплатить потребитель. Фактически это связано с отклонениями в технологическом процессе, т.е. возникает брак. Данное условие описывает потери при производстве дефектной продукции. Принимаем $u_n = u_{к3}$.

Приведенную выше совокупность условий и подусловий применяем на каждой операции, двигаясь против движения технологического процесса от последней n -й операции к первой,

формируя набор управлений $U_{1 \times n}$, приводящих систему из исходного состояния, в котором требования потребителя и возможности производителя не совпадают, в состояние полного соответствия. Помимо управляющей матрицы формируется набор требований $Q_{1 \times (n-1)}$ и $P_{1 \times (n-1)}$ к операции, находящейся выше исследуемой в технологической цепочки создания ценности. Таким образом, можно составить алгоритм формирования оптимального управления, а также исходных требований к собственным поставщикам. Данный алгоритм записан в виде блок-схем с использованием языка описаний (Specification and Description Language, SDL) на основе спецификаций ССИТТ.

В данном алгоритме использованы следующие переменные:

$R = (p \ q)$ - матрица типа real, составленная на основе требований клиента;

$$S = \begin{pmatrix} p_1 & q_1 \\ p_2 & q_2 \\ \dots & \dots \\ p_n & q_n \end{pmatrix} - \text{матрица исходного состоя-}$$

ния системы типа real, разбитая по операциям;

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \dots \\ u_n \end{pmatrix} - \text{матрица типа string, составлен-}$$

ная из оптимальных управлений на каждой операции;

n - переменная типа integer - количество операций (шагов) технологического процесса.

В алгоритме используются следующие постоянные константы:

$$u1 = \begin{pmatrix} Heijunka \\ SMED \end{pmatrix} = u_{к1} - \text{управление при}$$

обнаружении потери перепроизводства, массив типа string;

$$u2 = \begin{pmatrix} JIT \\ Kanban \end{pmatrix} = u_{к2} - \text{управление при об-}$$

наружении потери на излишних запасах, массив типа string;

$$u3 = \begin{pmatrix} Poka - Eke \\ СОК \\ Визуализация \\ Jidoka \end{pmatrix} = u_{k3} - \text{управление}$$

при обнаружении потери на дефектах, массив типа string;

$$u4 = \begin{pmatrix} ТРМ \\ Синхронизация \end{pmatrix} = u_{k4} - \text{управление}$$

при обнаружении потери на ожидания и простои, массив типа string;

$$u5 = \begin{pmatrix} Spagetti_chart \\ Сотовая_планировка \end{pmatrix} = u_{k5} - \text{уп-$$

равление при обнаружении потери на ожидания и простои, массив типа string;

$u6 = (\text{Устранить_операцию}) = u_{k6}$ - управление при обнаружении потери на ожидания и простои, массив типа string;

$u7 = (5S) = u_{k7}$ - управление при обнаружении неэффективности операций, массив типа string.

Перейдем к описанию алгоритма. На вход системы подаются переменные R , S и n . Перед началом "обнуляется" матрица управлений U . Вводятся внутренние переменные a , b , c , d , y , где a и b соответствуют требованиям клиента к n -й операции и принимают значения элементов матрицы R , а c и d соответствуют состоянию системы на n -й операции и принимают значения элементов матрицы S . Переменная y соответствует управлению на текущей операции. Предварительно y присваивается нулевое значение. Далее осуществ-

ляем проверку и формирование значения y в соответствии с условиями, приведенными в работе ранее. После окончания проверки заносим значение y в соответствующую ячейку матрицы управлений U . Все остальные внутренние переменные обнуляются.

После этого организуем цикл с переменной k , которая изменяется от n до 1 с шагом -1. Это означает перебор операций от предпоследней до первой в обратном порядке. Данный цикл аналогичен проверке условий с внутренними переменными, описанными выше. В результате алгоритма полностью заполняется матрица управлений U .

Таким образом, в результате нами был построен алгоритм оптимального устранения потерь. В дальнейшем алгоритм планируется включить в систему принятия решения об оптимальности устранения потерь.

¹ Савенков Д.Л. Практика внедрения "Бережливого производства" на промышленных предприятиях машиностроительного комплекса России. М., 2006.

² Imai Maasaki. Kaizen: The Key to Japan's Competitive Success. N.Y., 1986.

³ Мищенко А.В., Джамай Е.В. Динамическая задача определения оптимальной производственной программы // Менеджмент в России и за рубежом. 2002. ¹ 2.

⁴ Вэйдер М. Инструменты бережливого производства: Мини-руководство по внедрению методик бережливого производства. М., 2006.

⁵ Павлов В.Н. Математическая экономика. Новосибирск, 2000.

⁶ Черноморов Г.А. Теория принятия решений. Новочеркасск, 2002.

⁷ Сигео Синго. Изучение производственной системы Тойоты с точки зрения организации производства. М., 2006.

Поступила в редакцию 25.11.2009 г.