

© 2020 г. Б.В. КУПРИЯНОВ, канд. техн. наук, kuprianovb@mail.ru
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ОЦЕНКА И ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ РЕКУРСИВНОГО КОНВЕЙЕРА

Сформулировано понятие производительности рекурсивного конвейера, и задача оптимизации распределения возобновляемых ресурсов решена как задача максимизации производительности путем сведения к задаче целочисленного линейного программирования. Дано определение рекурсивных функций вычисления расписания процесса для некоторого произвольного распределения ресурсов. Результаты могут быть использованы при проектировании конвейера или для вычисления граничных оценок при использовании комбинаторных алгоритмов построения расписания.

Ключевые слова: теория расписаний, рекурсивные конвейеры, производительность конвейера, балансировка конвейера.

DOI: 10.31857/S0005231020050025

1. Введение

В массовом производстве [1] или при проектировании производственных линий распределение ресурсов (станков, инструментов, людей) осуществляется до начала производственного процесса. Так, проектирование сборочного конвейера является долгосрочным решением и обычно требует крупных капиталовложений. Задача расчета необходимого количества ресурсов и их распределения возникает и в тех случаях, когда необходимо выполнение некоторого объема заказа к заданному сроку. Вычисление оптимального распределения ресурсов (если это возможно) до построения расписания процесса позволяет существенно упростить задачу построения расписания процесса. В публикациях по теории расписаний большое количество работ посвящено данному вопросу. Существуют работы, которые фокусируются на методах оптимизации балансировки линии и планирования ресурсов на этапе проектирования сборочной линии [2]. Ghosh и Gagnon [3] делают комплексный обзор и анализ различных методов проектирования и планирования монтажа систем. Большая часть работ сконцентрирована на балансировке сборочного конвейера (ALB). Для ALB модели Хельгесоном (Helgeson) и соавторами было предложено использование методов поиска оптимальных решений, таких как линейное программирование [4, 5] и целочисленное программирование [6]. В [7] рассматривается задача распределения ресурсов как задача оптимального быстрогодействия. В настоящее время существует большое количество работ по оптимизации планирования ресурсов для традиционных задач теории расписаний, например [8–11].

В статье рассматриваются конвейеры, описываемые ациклическим ориентированным графом и набором рекурсивных функций [12]. Примеры при-

ложений таких конвейеров приводятся в [13]. Вводится понятие производительности конвейера и решается задача максимизации производительности на множестве распределений дискретных возобновляемых ресурсов. Конечный этап решения сводится к задаче целочисленного линейного программирования.

Статья имеет следующее содержание. Во второй главе дано формальное определение рекурсивного конвейера, приведен пример конвейера и временной диаграммы, описаны основные свойства конвейера. В третьей главе вводится определение производительности рекурсивного конвейера и формулируется задача оптимального распределения ресурсов как задача целочисленного программирования. Далее рассматривается пять этапов решения задачи. Приведены модифицированные рекурсивные функции. Каждый этап сопровождается примером. В четвертой главе рассматривается пять примеров решения задачи.

2. Определения

Модель конвейера, отношения временного предшествования которого определяются рекурсивными функциями (далее для краткости просто конвейера), представляет собой связный ациклический ориентированный граф $G = (V, A)$ с единственной конечной вершиной. V – множество вершин графа и n – количество вершин. A – множество дуг графа, упорядоченных пар вида (v, w) , где $v, w \in V$. Вершины графа помечены номерами от 1 до n таким образом, что первые n_0 ($1 \leq n_0 < n$) вершин являются начальными, а вершина n конечной. Каждая вершина графа обозначает либо производственную операцию, обрабатывающую заказ с помощью возобновляемого ресурса, либо спусковую функцию, определяющую отношения временного предшествования операций и не потребляющей ресурсы. Под операциями будем подразумевать как производственные операции, так и спусковые функции. Для идентификации типа вершины помечены с помощью отображения $type: V \rightarrow E$, где $E = \{\mathbf{bop}, \mathbf{op}, \mathbf{and}, \mathbf{mul}, \mathbf{red}, \mathbf{get1}, \mathbf{get2}, \mathbf{put}\}$ – конечное перечислимое множество. Каждая константа множества E обозначает тип вершины графа, а сама вершина имеет соответствующую графическую нотацию (см. рис. 1, а–ж).

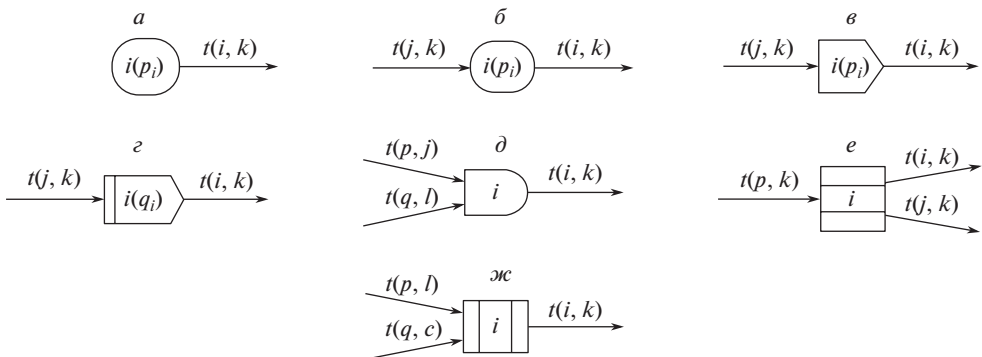


Рис. 1.

Из практики определено пять видов спусковых функций, имеющих применение для широкого круга конвейерных процессов. Однако можно определять новые исходя из конкретных потребностей. Для производственной операции и спусковых функций отношения предшествования описываются с помощью рекурсивных функций. Расписание выполнения операций строится с помощью суперпозиции рекурсивных функций, вычисляющих значение времени завершения обработки i -й операцией k -го заказа ($k \geq 0$). Для каждого элемента множества E своя функция, т.е. восемь функций и каждая вершина i графа в соответствии с ее типом ассоциируется с одной из этих функций. Рекурсивная функция $t(i, k)$ обеспечивает суперпозицию и является обращением к одной из них в соответствии с типом вершины.

$$(1) \quad t(i, k) = \begin{cases} \text{bop}(i, k), & \text{if } \text{type}(i) = \mathbf{bop}; \\ \text{op}(i, k), & \text{if } \text{type}(i) = \mathbf{op}; \\ \text{and}(i, k), & \text{if } \text{type}(i) = \mathbf{and}; \\ \text{mul}(i, k), & \text{if } \text{type}(i) = \mathbf{mul}; \\ \text{red}(i, k), & \text{if } \text{type}(i) = \mathbf{red}; \\ \text{get1}(i, k), & \text{if } \text{type}(i) = \mathbf{get1}; \\ \text{get2}(i, k), & \text{if } \text{type}(i) = \mathbf{get2}; \\ \text{put}(i, k), & \text{if } \text{type}(i) = \mathbf{put}. \end{cases}$$

Здесь имена рекурсивных функций для взаимно однозначного соответствия обозначаются теми же именами, что и типы вершин, но не жирным шрифтом.

Каждая вершина графа может иметь одну или две входных дуги в зависимости от типа вершины. Переход в процессе вычисления рекурсивной функции от вершины к вершине графа осуществляется с помощью следующих функций:

$\text{pred}(i)$ – определена для вершины, имеющей одну предшествующую вершину, и вычисляет номер j вершины графа такой, что существует единственная дуга $(v_j, v_i) \in A$;

(Следующие две функции определены для вершин, имеющих две предшествующие вершины. Условно будем считать: вершина 1 и вершина 2. Способ определения порядка в данном случае не имеет значения. Важно только, что он существует.)

$\text{pred1}(i)$ – вычисляет номер p 1-й вершины графа такой, что существует дуга $(v_p, v_i) \in A$;

$\text{pred2}(i)$ – вычисляет номер q 2-й вершины графа такой, что существует дуга $(v_q, v_i) \in A$.

Далее описываются все типы вершин: их назначение, тип и соответствующая типу рекурсивная функция. Все рекурсивные функции имеют два параметра: i – номер вершины графа и k – номер заказа. В каждый момент времени производственная операция может выполнять только один заказ и каждый ресурс может выполнять только одну операцию. Рекурсивные функции здесь сформулированы при условии, что каждой операции выделен один ресурс. Задача распределения ограниченного множества ресурсов и со-

ответствующие рекурсивные функции рассматриваются далее. Время начала функционирования конвейера полагается равным 0.

1. Начальная производственная операция (см. рис. 1,а) с номером $1 \leq i \leq n_0$ характеризуется временем выполнения p_i , которое является константой. Прерывание выполнения операции запрещено. Время завершения выполнения данной операцией k -го заказа определяется как время завершения выполнения $(k - 1)$ -го заказа плюс время выполнения операции p_i . Время выполнения нулевого заказа равно p_i .

$bop(i, k) :$

$$bop = \begin{cases} bop(i, k - 1) + p_i, & \text{if } k > 0, \\ p_i, & \text{if } k = 0. \end{cases}$$

2. Не начальная производственная операция (см. рис. 1,б) с номером $n_0 < i$ характеризуется временем выполнения p_i , которое является константой. Время завершения выполнения данной операцией k -го заказа определяется как максимум времен: завершения выполнения данной операцией $(k - 1)$ -го заказа и времени завершения выполнения предшествующей ей операцией k -го заказа, плюс время выполнения операции p_i . Время выполнения нулевого заказа равно времени выполнения нулевого заказа предшествующей операцией плюс p_i .

$op(i, k) :$

$$j = pred(i),$$

$$op = \begin{cases} \max(t(j, k), op(i, k - 1)) + p_i, & \text{if } k > 0, \\ t(j, k) + p_i, & \text{if } k = 0. \end{cases}$$

3. Спускная функция and (см. рис. 1,в) вычисляет время завершения выполнения k -го заказа для двух предшествующих операций p и q .

$and(i, k) :$

$$p = pred1(i), \quad q = pred2(i),$$

$$and = \max(t(p, k), t(q, k)).$$

4. Спускная функция мультиплицирования (см. рис. 1,г) для каждого времени завершения предшествующей ей операции j вычисляет q_i времен завершения функции i . Это означает, что на одно выполнение операции j осуществляется q_i выполнений операции, для которой функция i является предшествующей.

$mul(i, k) :$

$$j = pred(i), \quad l = \lfloor k/q_i \rfloor,$$

$$mul = t(j, l),$$

где $\lfloor x \rfloor$ обозначает целую часть x .

5. Спускоская функция редуцирования (см. рис. 1,*з*) является обратной по отношению к функции мультиплицирования, т.е. она вычисляет время завершения выполнения партии из q_i заказов.

$red(i, k) :$

$$\begin{aligned} j &= pred(i), \quad l = (k + 1)q_i - 1, \\ red &= t(j, l). \end{aligned}$$

6. Спускоская функция раздачи (см. рис. 1,*е*) имитирует разделение конвейерных операций на два потока. Для пользователя она выступает как одна функция, а в реализации она разбивается на две (условно верхнюю и нижнюю). Для верхнего варианта спускоская функция вычисляет времена завершения четных заказов, т.е. $t(i, 0) = t(p, 0)$, $t(i, 1) = t(p, 2)$, $t(i, 2) = t(p, 4)$, \dots

$get1(i, k) :$

$$\begin{aligned} p &= pred(i), \quad l = 2k, \\ get1 &= t(p, l). \end{aligned}$$

Для нижнего вычисляет времена завершения выполнения нечетных заказов, т.е. $t(j, 0) = t(p, 1)$, $t(j, 1) = t(p, 3)$, $t(j, 2) = t(p, 5)$, \dots

$get2(i, k) :$

$$\begin{aligned} p &= pred(i), \quad l = 2k + 1, \\ get2 &= t(p, l). \end{aligned}$$

7. Спускоская функция приема (см. рис. 1,*ж*) является обратной по отношению к функции раздачи и имитирует слияние двух конвейеров в один. Проще всего это пояснить последовательностью значений $t(i, k)$:

$$t(i, 0) = t(p, 0), \quad t(i, 1) = t(q, 0), \quad t(i, 2) = t(p, 1), \quad t(i, 3) = t(q, 1), \quad t(i, 4) = t(p, 2), \quad t(i, 5) = t(q, 2), \dots$$

$put(i, k) :$

$$\begin{aligned} p &= pred1(i), \quad l = k/2, \\ q &= pred2(i), \quad c = (k - 1)/2, \\ put &= \begin{cases} t(p, l), & \text{if } k = 0, \\ \max(put(i, k - 1), t(q, c)), & \text{if } k \bmod 2 = 1, \\ \max(put(i, k - 1), t(p, l)), & \text{if } k \bmod 2 = 0, \end{cases} \end{aligned}$$

где $x \bmod y$ – остаток от деления x на y .

Вычисление времени завершения выполнения k -го заказа конвейером осуществляется с помощью обращения к рекурсивной функции $t(n, k)$ по формуле (1). Эта же формула вычисляет время завершения выполнения k -го заказа любой i -й операцией конвейера обращением $t(i, k)$. Рекурсивность в данном случае выступает в двух видах:

— значение функции для некоторого k зависит от значения этой функции для $(k - 1)$ или другого меньшего значения;

— в случае суперпозиций функции она может обращаться к самой себе с другим значением аргумента (см. в разделе 4 пример 1).

Важно понимать, что рекурсивные функции определяют отношение временного предшествования и используются в дальнейшем только для вычисления расписания выполнения операций. Никакого отношения к управлению конвейером они не имеют. Управление конвейером аналогично классическим конвейерам, может осуществляться множеством способов и зависит от технологического решения. В разделе 4 в примере 1 показана модель конвейера, соответствующая ему суперпозиция рекурсивных функций и вычисление такой функции для некоторых значений аргументов.

Временная сложность вычисления расписания с помощью таких функций — $O(nk)$, где n — количество вершин графа, k — количество обработанных заказов.

Интервалом обработки k -го заказа i -й операцией конвейера $d(i, k)$ $1 \leq n$, $k \geq 1$ назовем продолжительность времени между моментами завершения i -й операции при обработке k и $(k - 1)$ заказов:

$$d(i, k) = t(i, k) - t(i, k - 1).$$

Очевидно, что $d(i, k) \geq p_i$. Величины $d(n, k)$ для последовательности значений k определяют промежутки времени, через которые с конвейера сходит продукция. В общем случае интервал операции зависит от номера заказа k . Для описанных выше функций предшествования в [14] показано, что в общем случае интервал операции i конвейера для первых ks_i заказов меняется по некоторому закону переходного процесса, далее процесс переходит в стационарный режим и интервал меняется по закону периодической функции. Каждую операцию i конвейера можно характеризовать следующим кортежем:

$$(t0_i, ks_i, ts_i, D_i, T_i),$$

где i — номер операции;

$t0_i$ — время завершения операции i при обработке нулевого заказа (при i , равном n , это будет время выхода первого изделия с конвейера);

ks_i — номер заказа, начиная с которого операция i переходит в стационарный режим выполнения;

ts_i — время, начиная с которого операция i переходит в стационарный режим выполнения;

D_i — определяется как сумма интервалов операции i за период;

T_i — период колебания интервала операции i , измеряется в количествах заказов и является безразмерной величиной.

Для стационарного режима можно ввести понятие направляющей времени завершения выполнения заказа $t(i, k)$ операции i — луча, выходящего из точки (ks_i, ts_i) на котором лежат точки

$$t(i, k) = ts_i + kD_i, \quad k = 0, T_i, 2T_i, \dots$$

Параметры $t_{0i}, ks_i, ts_i, D_i, T_i$ являются константами, и в случае рассмотрения конвейера ($i = n$) для удобства будем обозначать их как t_0, k_s, t_s, D, T . В определении не уточняется, относится ли определение стационарного режима работы конвейера ко всем операциям или только к завершающей операции конвейера. В данной статье будем считать, что это относится ко всем операциям конвейера, т.е. после обработки заказа k_s все операции конвейера перешли в стационарный режим выполнения.

3. Решение задачи оптимизации

В данном разделе рассматривается решение задачи оптимизации распределения ресурсов для стационарного режима работы конвейера как задачи целочисленного линейного программирования.

Введем ряд обозначений для ресурсов:

J – количество множеств ресурсов ($1 \leq j \leq J$). Каждое множество содержит ресурсы одного типа;

r_j – количество ресурсов в j -м множестве. Ресурс может иметь составную структуру, например, для выполнения операции нужен один станок и два человека. Набор ресурсов, обслуживающий одну операцию, будем называть комплектом и с каждой операцией связывать некоторое целое число комплектов;

a_{ij} – коэффициент комплектации i -й операции j -м ресурсом, например, если выполнение i -й операции требует наличия одного станка и трех рабочих, то $a_{i1} = 1$, $a_{i2} = 3$;

x_i – количество комплектов ресурсов, используемых i -й операцией, является целым числом без размерности. Если i -й операции поставлено в соответствии x_i комплектов, то операция может x_i раз выполняться с совмещением во времени;

W_i – производительность i -й операции конвейера, которая измеряется количеством выполнений операции в составе конвейера в единицу времени и является вещественным числом.

Интервал конвейера $d(n, k)$ определяет его производительность. Чем больше интервал конвейера, тем меньше его производительность. Каждая операция i конвейера [15] может быть охарактеризована кратностью выполнения ω_i . Данная величина показывает, сколько раз операция i выполняется в пересчете на одно выполнение конечной операции конвейера. Иначе, сколько раз выполняется операция i при производстве одного изделия конвейера. Кратность операции увеличивает время выполнения операции в ω_i раз и соответственно уменьшает производительность операции в ω_i раз. В общем случае производительность является величиной переменной и может вычисляться по формуле

$$W_i(k) = g_i \frac{x_i}{d(i, k)\omega_i},$$

т.е. прямо пропорционально зависит от количества комплектов ресурсов x_i и обратно пропорционально зависит от произведения интервала опера-

ции $d(i, k)$ на ее кратность ω_i , g_i является коэффициентом пропорциональности. В случае стационарного процесса можно рассматривать среднюю производительность (не зависящую от k) за период колебания интервала

$$W_i = g_i \frac{T_i x_i}{D_i \omega_i},$$

где D_i – приращение интервала за период, T_i – величина периода и $\frac{D_i}{T_i}$ – средний интервал i -й операции за период, для которого справедливо неравенство $\frac{D_i}{T_i} \geq p_i$. Далее в статье рассматривается только средняя производительность в стационарном режиме, которую для краткости будем называть производительностью. В связи с последним неравенством введем неравенство для производительности

$$W_i \leq g_i \frac{x_i}{p_i \omega_i}.$$

Поскольку p_i , g_i и ω_i являются константами, для упрощения введем новую константу c_i , которая вычисляется по формуле

$$c_i = \frac{g_i}{p_i \omega_i}$$

и является вещественным числом. В этом случае для производительности справедливо неравенство

$$W_i \leq c_i x_i.$$

В терминах введенных определений задача формулируется следующим образом. Пусть имеется конвейер, выполняющий n операций, каждая продолжительностью $p_i \geq 0$, $1 \leq i \leq n$. Имеется некоторое множество ресурсов, разбитое на J групп по r_j , $1 \leq j \leq J$ в каждой группе. В этом случае задача оптимального распределения ресурсов является задачей целочисленного линейного программирования с целевой функцией

$$W \xrightarrow{x_i} \max$$

и системой ограничений

$$\sum_{i=1}^n a_{ij} x_i \leq r_j, \quad 1 \leq i \leq n, \quad 1 \leq j \leq J,$$

$$W_i \leq c_i x_i, \quad 1 \leq i \leq n,$$

$$W \leq W_i, \quad 1 \leq i \leq n,$$

$$1 \leq x_i, \quad 1 \leq i \leq n.$$

Здесь W – производительность конвейера. Ограничение $x_i \geq 1$ обусловлено тем, что при $x_i = 0$ конвейер не может функционировать. Решение задачи разбивается на 5 этапов, от модели в виде цепочки операций, далее путем

последовательного рассмотрения спусковых функций к решению модели общего вида.

Эман 1. Рассмотрим конвейер как цепочку из n производственных операций. Известно, что производительность такого конвейера определяется его узким местом, т.е. операцией с самым большим временем выполнения. С учетом данного свойства максимум производительности достигается максимизацией минимальной производительности операций конвейера. Решим эту задачу методом линейного программирования. Целевая функция W – производительность конвейера, которая равна минимальной производительности операций конвейера, т.е.

$$W = \min_{1 \leq i \leq n} W_i.$$

Для каждого ресурса можно составить неравенство ограничения на ресурс

$$\sum_{i=1}^n a_{ij}x_i \leq r_j, \quad 1 \leq i \leq n, \quad 1 \leq j \leq J.$$

Для производительности W_i i -й операции конвейера выполняется неравенство

$$W_i \leq c_i x_i, \quad 1 \leq i \leq n.$$

Запишем эту постановку задачи формально в виде задачи линейного программирования

$$\begin{aligned} W &\xrightarrow{x_i} \max, \\ W &\leq W_i, \\ \sum_{i=1}^n a_{ij}x_i &\leq r_j, \quad 1 \leq i \leq n, \quad 1 \leq j \leq J, \\ W_i &\leq c_i x_i, \quad 1 \leq i \leq n, \\ 1 &\leq x_i, \quad 1 \leq i \leq n. \end{aligned}$$

Все переменные x_i принимают целочисленные значения, W может быть вещественным. Таким образом, задача является разновидностью задачи целочисленного линейного программирования. Описанный метод рассмотрен в примере 3 раздела 4.

Модифицируем рекурсивные функции для случая, когда i -я операция располагает x_i комплектами ресурсов. Основу нового вида рекурсивных функций определяет тот факт, что если i -я операция обеспечена x_i количеством комплектов, то x_i выполнений этой операции могут пересекаться во времени. Для начальной операции (см. рис. 1,а) новые рекурсивные функции определяются следующим образом:

$$\begin{aligned} t(i, k) &= p_i, \quad \text{если } 1 \leq i \leq n_0 \text{ и } 0 \leq k < x_i; \\ t(i, k) &= t(i, k - x_i) + p_i, \quad \text{если } 1 \leq i \leq n_0 \text{ и } k \geq x_i. \end{aligned}$$

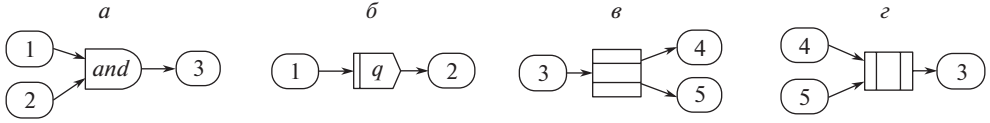


Рис. 2.

Если операция не начальная (см. рис. 1,б), то рекурсивные функции имеют вид

$$\begin{aligned}
 t(i, k) &= t(j, k) + p_i, & \text{если } i > n_0 \text{ и } 0 \leq k < x_i; \\
 t(i, k) &= \max(t(j, k), t(i, k - x_i)) + p_i, & \text{если } i > n_0 \text{ и } k \geq x_i.
 \end{aligned}$$

Изменение расписания при использовании модифицированных рекурсивных функций хорошо видно на временных диаграммах в системе координат. По оси абсцисс отложим номер заказа, а по оси ординат – время. Время выполнения i -й операции при обработке k -го заказа будет задаваться отрезком с координатами $(t(i, k) - p_i, t(i, k))$ и отмечаться номером операции. В примере 2 на рис. 5,б,в приведены временные диаграммы для рассматриваемого конвейера.

Этап 2. В том случае, когда в модели конвейера используется спусковая функция mul , она изменяет кратность ω_i выполнения следующих за ней операций. В этом случае кратности выполнения некоторых операций будут отличны от единицы. Это влечет за собой изменение коэффициентов c_i в соответствии с формулой (2). В остальном задача остается без изменений. В примере 5 рассматривается конвейер со спусковой функцией mul , решение задачи и соответствующие временные диаграммы.

Этап 3. Из [15] известно, что спусковые функции: редукции, раздачи и приема, по сути как и операция мультиплицирования, осуществляют либо увеличение кратности операции в два раза (get), либо уменьшение (red, put). Поэтому для данных спусковых функций метод решения будет аналогичным. Так, для моделей на рис. 2,б,в,г выполняются соотношения

$$\omega_1 = q\omega_2, \quad \omega_3 = 2\omega_4 = 2\omega_5, \quad \omega_8 = 2\omega_6 = 2\omega_7.$$

Для данных соотношений и существующих ресурсных ограничений решается задача максимизации для моделей, содержащих данные спусковые функции. Ограничения по ресурсам специфики не имеют, а ограничения по производительности имеют следующий вид:

для варианта а)

$$\begin{aligned}
 W &\leq c_1 x_1; \\
 W &\leq c_2 x_2,
 \end{aligned}$$

для варианта б)

$$\begin{aligned}
 W &\leq c_3 x_3; \\
 W &\leq c_4 x_4; \\
 W &\leq c_5 x_5,
 \end{aligned}$$

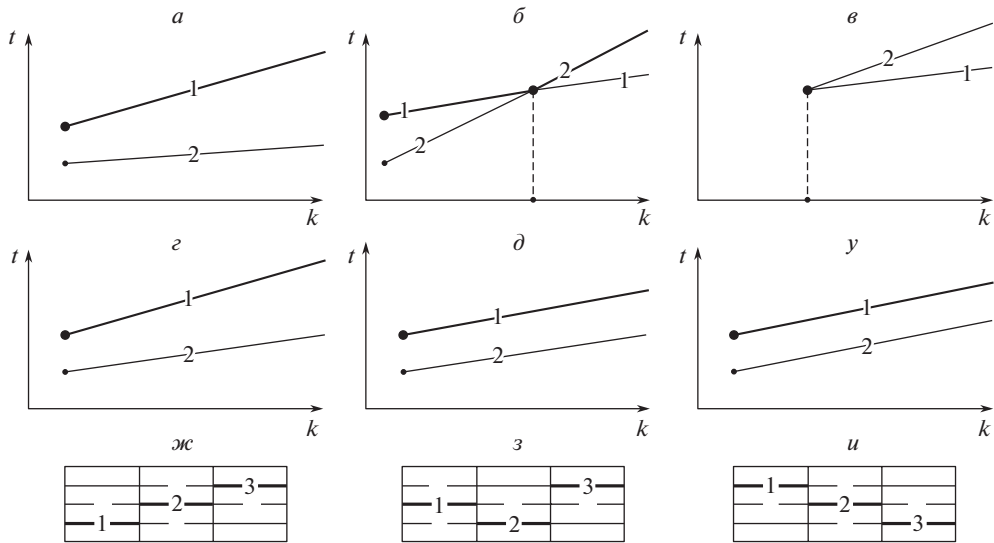


Рис. 3.

для варианта в)

$$W \leq c_6 x_6;$$

$$W \leq c_7 x_7;$$

$$W \leq c_8 x_8.$$

В примере 7 используются все спусковые функции, в том числе и данные.

Этап 4. Рассмотрим спусковую функцию *and*. На рис. 2,а представлена конструкция конвейера, в которой 1, 2 и 3 будем интерпретировать как некоторые конвейеры. Поскольку рассматривается стационарный конвейерный процесс и определены соответствующие направляющие функции $t(i, k)$, то на рис. 3,а,б,в отобразим направляющие для двух конвейеров 1–2 и функции *and*. График изменения направляющей спусковой функции *and* выделен жирной линией и в данном случае совпадает с графиком 1 (в соответствии с определением рекурсивной функции для *and*). Направляющие конвейеров 1 и 2 могут выходить из одной точки, быть параллельны или направляющая 2 может быть выше направляющей 1, но они не могут пересекаться. Пересечение направляющих 1 и 2, как показано на рис. 3,б, означает, что переходный процесс завершился в точке с абсциссой $k = k_s$. Следовательно ситуация сводится к случаю, когда обе направляющие выходят из одной точки рис. 3,в. Таким образом ситуация, показанная на рис. 3,а, является основной, а остальные сводятся к ней. Проанализируем данную ситуацию. Так как $t(i, k)$ функции *and* определяется $t(j, k)$ конвейера, который имеет меньшую производительность (в данном случае конвейер 1), то существующие ресурсы, если они есть, необходимо выделить конвейеру 1. На рис. 3,г показаны исходные графики, после выделения ресурсов конвейеру 1 его производительность вырастет и направляющая станет более пологой, как показано на рис. 3,д. Если ресурсы заимствованы не из свободного резерва, а у конвейера 2, то

производительность конвейера 2 упадет и его направляющая станет более крутой (см. рис. 3, *д*). Данные примеры и рассуждения показывают, что распределение ресурсов между конвейерами 1 и 2 должны быть такими, чтобы их производительности были равны и, как следствие, направляющие были параллельны (см. рис. 3, *е*).

Если после данного предела в какой-либо конвейер добавить ресурс, то направляющие пересекутся и ситуация вернется к исходной. Более того, если рассматривать соотношения производительностей трех конвейеров, то возможны три существенных варианта, представленные на рис. 3, *ж, з, и*. В варианте *ж* производительность конвейера определяется конвейером 1, соотношение двух других не имеет значения. В варианте *з* аналогичным образом производительность определяется конвейером 2, а в варианте *и* – конвейером 3.

Окончательным итогом данных рассуждений является равенство трех производительностей $W_1 = W_2 = W_3$ как условие оптимальности, а система ограничений на производительности выглядит следующим образом:

$$\begin{aligned} W &\longrightarrow \max; \\ W &\leq W_1; \\ W &\leq W_2; \\ W &\leq W_3. \end{aligned}$$

Этап 5. Прежде чем рассмотреть заключительный этап, сформулируем теорему.

Теорема 1. Если рекурсивный конвейер, состоит из n описанных выше операций и спусковых функций и W – производительность конвейера, то для любых двух операций $i \neq j$ ($1 \leq i, j \leq n$) условие $W \longrightarrow \max$ означает, что

$$\left| \frac{x_i}{p_i \omega_i} - \frac{x_j}{p_j \omega_j} \right| \longrightarrow \min.$$

Доказательство теоремы фактически было рассмотрено выше для всех видов спусковых функций и операции. Данная теорема играет важную роль при формулировании на основании графической модели конвейера системы неравенств соответствующей задачи линейного программирования. Описание системы неравенств, относящейся к ограничениям на ресурсы, не представляет проблемы. На основании данной теоремы система неравенств для производительности будет выглядеть следующим образом:

$$W \leq \frac{x_i}{p_i \omega_i},$$

где i ($1 \leq i \leq n$) является номером производственной операции (не спусковой функции). Поставленная задача полностью решена. Для иллюстрации метода в примере 5 рассматривается оптимизация распределения ресурсов для конвейера, содержащего производственные операции и все описанные спусковые функции.

4. Примеры решения задач

Пример 1. Рассмотрим пример конвейера, представленный на рис. 4,а, на рис. 4,б представлена соответствующая конвейеру суперпозиция рекурсивных функций, вычисляющая время завершения конвейером выполнения k -го заказа.

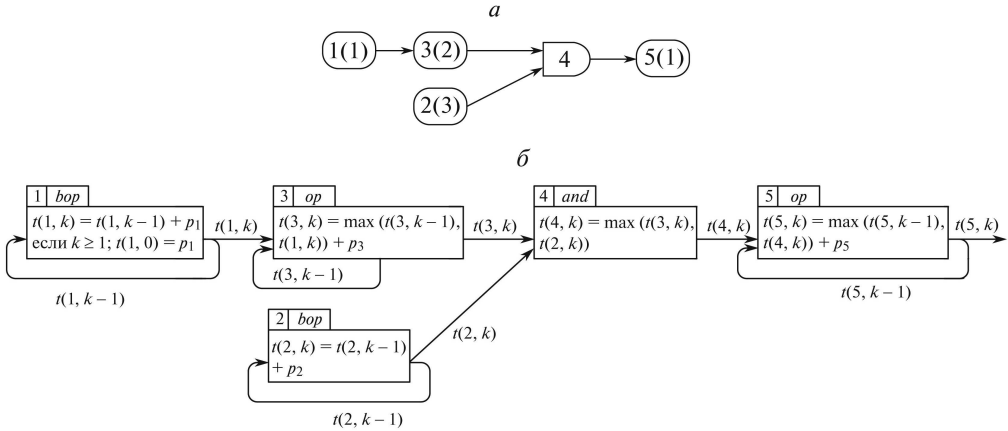


Рис. 4.

Ниже приведен пример вычисления времени завершения выполнения 5-й операцией 2-го заказа $t(5, 2)$ для представленной модели. Для каждой операции – одна исполнительная машина. Другие параметры: $n = 5$, $n_0 = 2$. Приведем последовательность вычисления рекурсивных функций:

$$\begin{aligned}
 t(5, 2) &= \max(t(4, 2), t(5, 1)) + 1; \\
 t(5, 1) &= \max(t(4, 1) + t(5, 0)) + 1; \\
 t(5, 0) &= t(4, 0) + 1; \\
 t(4, 2) &= \max(t(2, 2), t(3, 2)); \\
 t(4, 1) &= \max(t(2, 1), t(3, 1)); \\
 t(4, 0) &= \max(t(2, 0), t(3, 0)); \\
 t(3, 2) &= \max(t(1, 2), t(3, 1)) + 2; \\
 t(3, 1) &= \max(t(1, 1), t(3, 0)) + 2; \\
 t(3, 0) &= t(1, 0) + 2; \\
 t(2, 2) &= t(2, 1) + 4; \\
 t(2, 1) &= t(2, 0) + 4; \\
 t(2, 0) &= 4; \\
 t(1, 2) &= t(1, 1) + 1; \\
 t(1, 1) &= t(1, 0) + 1; \\
 t(1, 0) &= 1.
 \end{aligned}$$

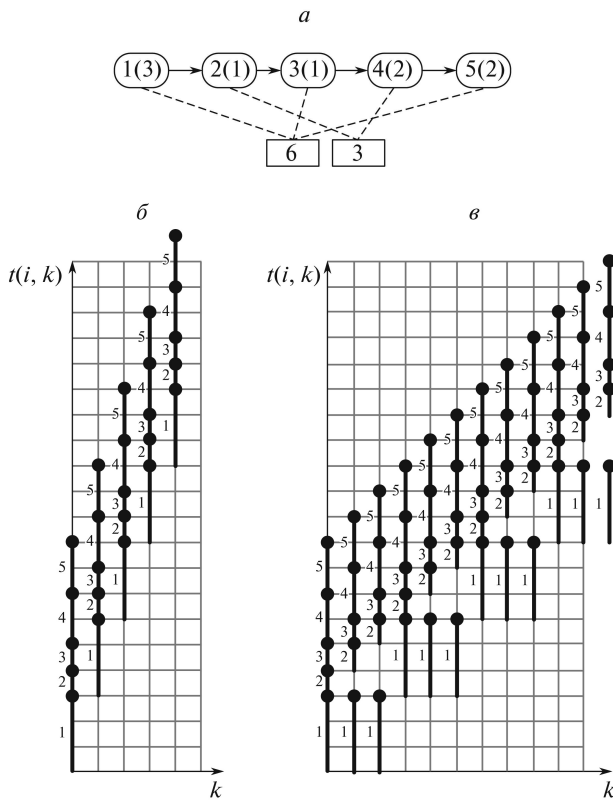


Рис. 5.

Пример 2. Пусть у нас имеется конвейер в виде цепочки из пяти операций (см. рис. 5). Времена выполнения операций указаны в скобках. Имеется два комплекта ресурсов. Первый – 6 единиц, второй – 3. Использование ресурсов операциями указаны на рисунке пунктирными линиями. В комплекты операций входит не больше чем по одному ресурсу (т.е. все $0 \leq a_{ij} \leq 1$). Целевая функция $W \rightarrow \max$. Система ограничений будет выглядеть следующим образом:

а) ограничения на ресурсы $\begin{cases} x_1 + x_3 + x_5 \leq 6; \\ x_2 + x_4 \leq 3; \end{cases}$

б) поскольку для решения задачи оптимизации представляют интерес не абсолютные значения производительностей, а относительные, коэффициенты производительности c_i для удобства преобразуем в целые числа по формуле

$$c_i = \frac{lcm(c_1, c_2, \dots, c_5)}{c_i},$$

где lcm – наименьшее общее кратное. Осуществляя данное преобразование c_i , сохраняем пропорции между коэффициентами, но преобразуем их в целые числа. Вычисленные таким образом коэффициенты представлены в табл. 1.

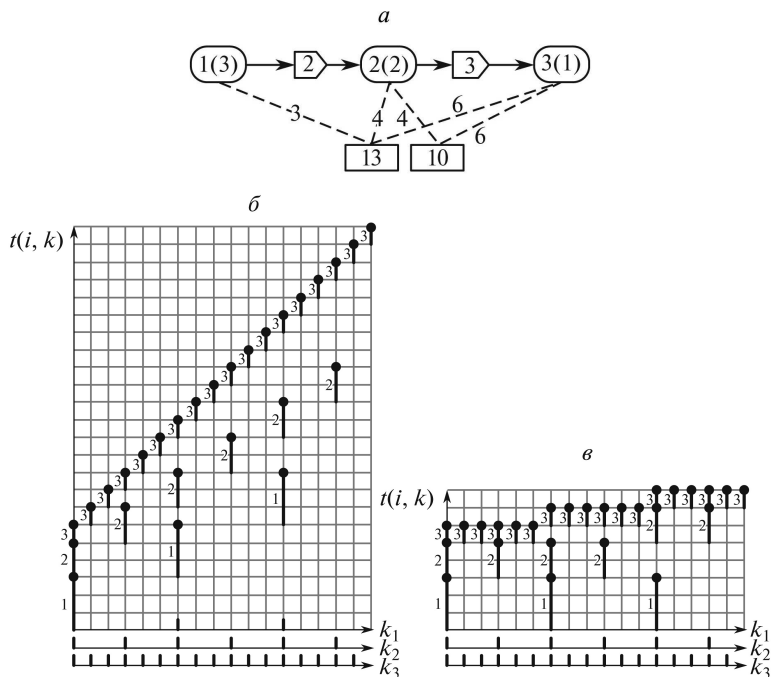


Рис. 6.

Ограничения на производительность будут выглядеть следующим образом:

$$\begin{aligned}
 W &\leq 2x_1; \\
 W &\leq 6x_2; \\
 W &\leq 6x_3; \\
 W &\leq 3x_4; \\
 W &\leq 3x_5.
 \end{aligned}$$

Решение данной системы дает следующий результат:

$$x_1 = 3, \quad x_2 = 1, \quad x_3 = 1, \quad x_4 = 2, \quad x_5 = 2, \quad W = 6.$$

Построенная на основании вычисленного распределения ресурсов и новых рекурсивных функций временная диаграмма обработки двенадцати заказов будет иметь вид, представленный на рис. 5,б. На рис. 5,б представлена диаграмма для случая, когда каждая операция использует по одному комплекту ресурсов.

Пример 3. Рассмотрим пример конвейера рис. 6,а. В данном случае присутствуют спусковые функции t_{ul} и кратности выполнения операций будут отличны от единицы.

Таблица 1. Параметры линейного конвейера

Номер операции i	1	2	3	4	5
Время выполнения операции t_i	3	1	1	2	2
Коэффициент производительности c_i	1/3	1	1	1/2	1/2
Приведенный коэффициент производительности c_i	2	6	6	3	3

Таблица 2. Параметры конвейера с функцией мультиплицирования

Номер операции i	1	2	3
Время выполнения операции t_i	9	4	1
Кратность операции ω_i	1/6	1/3	1
Приведенная кратность ω_i	1	2	6
Коэффициент производительности c_i	1/9	1/8	1/6
Приведенный коэффициент производительности c_i	8	9	12

В табл. 2 представлены параметры модели. В общем случае кратности являются дробными величинами, и перейти к целым приведенным значениям можно по формуле

$$\omega_i = \frac{lcm(\omega_1, \omega_2, \dots, \omega_5)}{\omega_i'}$$

где ω_i' – исходная кратность операции. Задача линейного программирования будет иметь следующий вид:

$$\begin{aligned} W &\longrightarrow \max; \\ \begin{cases} x_1 + x_2 + x_3 \leq 13; \\ x_2 + x_3 \leq 10; \end{cases} \\ W &\leq 4x_1; \\ W &\leq 3x_2; \\ W &\leq 2x_3. \end{aligned}$$

Решение данной системы дает следующий результат:

$$x_1 = 3, x_2 = 4, x_3 = 6, W = 12.$$

Рекурсивные функции вычисления $t(i, k)$ остаются прежними. На рис. 6, б, в показаны две диаграммы: вариант диаграммы б, когда каждая операция имеет один комплект ресурсов, и вариант в, когда ресурсы распределены в соответствии с найденным решением. В общем случае каждая операция должна отображаться на отдельной временной диаграмме, однако для наглядности их имеет смысл совмещать, при этом для каждой операции рисовать свою ось абсцисс.

Пример 4. Рассмотрим пример модели, представленной на рис. 7, а. Из предыдущих рассуждений следует, что операции 1, 2 должны иметь равную производительность, а так как функция *and* включена последовательно с операцией 3, то они все должны быть равны между собой для выполнения условия оптимальности. Производительность конвейера должна быть максимизирована. Параметры модели представлены в табл. 3. Целевая функция $W \longrightarrow \max$. Система ограничений будет выглядеть следующим образом:

$$\text{а) ограничения на ресурсы } \begin{cases} x_1 + x_2 + x_3 \leq 5; \\ x_2 + x_3 \leq 2; \end{cases}$$

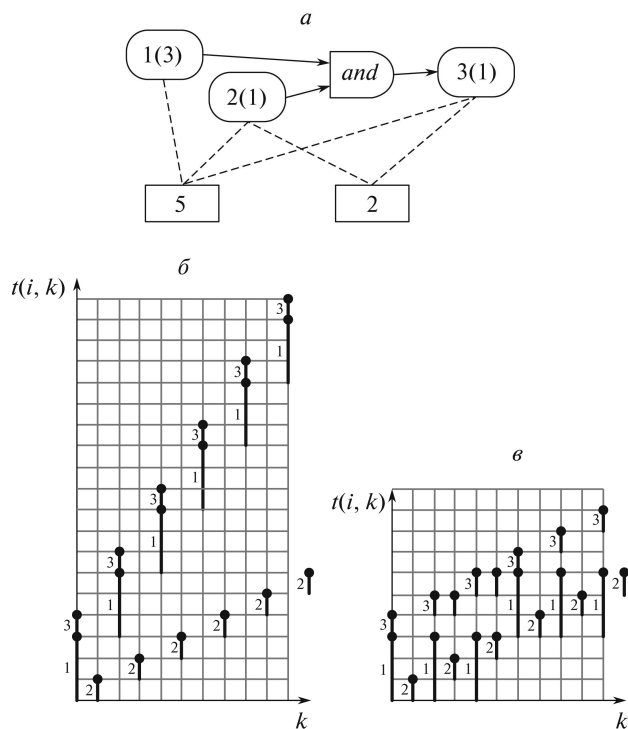


Рис. 7.

б) ограничения на производительность

$$\begin{aligned}
 W &\leq x_1; \\
 W &\leq 3x_2; \\
 W &\leq 3x_3.
 \end{aligned}$$

Решение данной системы дает следующий результат:

$$x_1 = 3, x_2 = 1, x_3 = 1, W = 3.$$

На рис. 7,б,в показаны диаграммы выполнения конвейера *a* – с одним комплектом на каждую операцию, а *б* – в соответствии с вычисленным распределением ресурсов.

Ось абсцисс у всех операций общая. Отрезки, соответствующие операции 2, смещены вправо от своих делений. Это сделано для того, чтобы не сливались диаграммы разных операций.

Таблица 3. Параметры конвейера с функцией and

Номер операции i	1	2	3
Время выполнения операции p_i	3	1	1
Кратность операции ω_i	1	1	1
Приведенный коэффициент производительности c_i	1	3	3

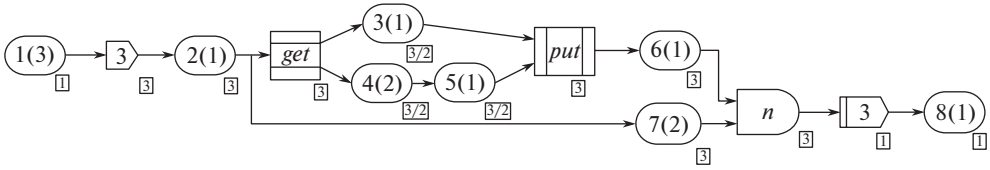


Рис. 8.

Пример 5. На рис. 8 представлен пример конвейера, использующий все рассмотренные спусковые функции. Справа снизу у каждой операции представлена неприведенная кратность ее выполнения. Целевая функция определяется как $W \rightarrow \max$. Используется три множества ресурсов. В первом 44 единицы, во втором 26 и в третьем 12. Каждый комплект может содержать только по одному ресурсу. В табл. 4 представлены характеристики конвейера и матрица использования ресурсов операциями конвейера. Система ограничений на ресурсы выглядит следующим образом

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \leq 44; \\ x_1 + x_2 + x_7 + x_8 \leq 26; \\ x_3 + x_4 + x_5 \leq 12. \end{cases}$$

Ограничения на производительность в соответствии с теоремой следующие:

$$\begin{aligned} W &\leq c_1 x_1; \\ W &\leq c_2 x_2; \\ W &\leq c_3 x_3; \\ W &\leq c_4 x_4; \\ W &\leq c_5 x_5; \\ W &\leq c_6 x_6; \\ W &\leq c_7 x_7; \\ W &\leq c_8 x_8. \end{aligned}$$

Решение задачи имеет следующий вид:

$$x_1 = 6, x_2 = 6, x_3 = 3, x_4 = 6, x_5 = 3, x_6 = 6, x_7 = 12, x_8 = 2, W = 12.$$

Таблица 4. Характеристики конвейера и матрица использования ресурсов

Номер операции i	1	2	3	4	5	6	7	8
Время выполнения операции p_i	3	1	1	2	1	1	2	1
Приведенная кратность операции ω_i	2	6	3	3	3	6	6	2
Приведенный коэффициент производительности c_i	2	2	4	2	4	2	1	6
Использование 1-го ресурса	1	1	1	1	1	1	1	1
Использование 2-го ресурса	1	1	0	0	0	0	1	1
Использование 3-го ресурса	0	0	1	1	1	0	0	0

5. Заключение

Разработанная в статье методика решения задачи распределения ресурсов позволяет в автоматическом режиме на основе графической модели конвейера и матрицы использования операциями ресурсов формировать исходные данные для задачи целочисленного линейного программирования. Решение данной задачи позволяет вычислить верхнюю оценку времени изготовления k изделий как частное от деления числа изделий на производительность конвейера при условии вычисления параметров в реальных величинах. Полученные результаты могут быть использованы не только при непосредственном проектировании конвейера, но и осуществлять перераспределение ресурсов при настройке конвейера на новый заказ. Еще одно возможное применение метода – получение граничных оценок для других, более сложных и эффективных методов оптимизации в теории расписаний. Решение задачи в данной постановке открывает перспективу для решения соответствующей двойственной задачи линейного программирования. Следует отметить, что полный список спусковых функций, приведенный в [12, 14], включает дизъюнкцию *or*. Однако для данной функции отсутствует общее решение задачи оптимизации распределения ресурсов. Поэтому она в данной статье не рассматривается.

СПИСОК ЛИТЕРАТУРЫ

1. Сачко Н.С. Организация и оперативное управление машиностроительным производством. Минск: ООО «Новое знание», 2005.
2. Rekiek B., Dolgui A., Delchambre A., Bratcu A. State of art of optimization methods for assembly line design // Ann. Rev. Control. 2002. No. 26. С. 163–174.
3. Ghosh S., Gagnon R.J. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems // Int. J. Product. Res. 1989. V. C-27. P. 637–670.
4. Helgeson W.B., Salvesson M.E., Smith W.W. How to balance an assembly line // Techni. Report, Carr Press. 1954. New Caraan, Conn.
5. Bowman E.H. Assembly line balancing by linear programming // Oper. Res. 1960. No. 8(3). P. 385–389.
6. Salvesson M.E. The assembly line balancing problem // J. Indust. Engineer. 1955. No. 6. P. 18–25.
7. Бурков В.Н. Распределение ресурсов как задача оптимального быстрогодействия // АиТ 1966. Т. 27. Вып. 7. С. 119–129.
8. Ranjbar M., Kianfar F., Shadrokh S. Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm // Appl. Math. Comput. 2008. Т. 196. Вып. 2. С. 879–888.
9. Kaabi J., Harrath Y. A survey of parallel machine scheduling under availability constraints // Int. J. Comput. Inform. Technol. 2014. Т. 3. Is. 2. P. 238–245.
10. Боброва Е.А., Сервах В.В. Построение циклических расписаний при наличии параллельных машин // Дискретный анализ и исследование операций. 2017. № 1. Т. 24. С. 5–20.
11. Мезенцев Ю.А., Эстрайх И.В. Эффективный параметрический алгоритм оптимизации расписаний параллельных систем с заданным расписанием начала обслуживания // Научн. вест. НГТУ. 2018. Т. 72. № 3. С. 87–106.

12. *Куприянов Б.В.* Рекурсивные конвейерные процессы – основные свойства и характеристики // Экономика, статистика и информатика. Вестн. УМО. 2015. № 1. С. 163–170.
13. *Куприянов Б.В.* Применение модели конвейерных процессов рекурсивного типа для решения прикладных задач // Экономика, статистика и информатика. Вестн. УМО. 2014. № 5. С. 170–179.
14. *Куприянов Б.В.* Метод эффективного анализа модели рекурсивного конвейерного процесса // АиТ. 2017. № 3. С. 63–79.
Kupriyanov B.V. Method of Efficient Analysis of the Recursive Conveyor Process Models // Autom. Remote Control. 2017. V. 78. No. 3. С. 435–449.
15. *Куприянов Б.В.* Вычисление некоторых производственных характеристик рекурсивного конвейера // Открытое образование. 2016. № 1. Т. 20. С. 11–16.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 17.07.2019

После доработки 23.10.2019

Принята к публикации 28.11.2019