

К.В. КРОТОВ

**МНОГОУРОВНЕВАЯ МОДЕЛЬ ПОСТРОЕНИЯ РАСПИСАНИЙ
ОБРАБОТКИ ПАРТИЙ ДАННЫХ В КОНВЕЙЕРНЫХ
СИСТЕМАХ ПРИ ФОРМИРОВАНИИ КОМПЛЕКТОВ И
НАЛИЧИИ ОГРАНИЧЕНИЙ**

Кротов К.В. Многоуровневая модель построения расписаний обработки партий данных в конвейерных системах при формировании комплектов и наличии ограничений.

Аннотация. Рассматривается задача построения расписаний обработки партий данных различных типов в конвейерной системе при наличии ограничений на время ее функционирования и условия формирования комплектов из результатов обработки. В статье исследуется первый этап решения указанной задачи, связанный с обоснованием многоуровневой модели принятия решений по составам партий данных, составам групп партий, обрабатываемых в течение заданных интервалов времени функционирования системы, расписаний обработки партий данных в группах с учетом условия формирования из результатов обработки комплектов разных типов.

Ключевые слова: модель многоуровневого программирования, партии данных, группы партий данных, комплекты, формируемые из результатов обработки.

Krotov K.V. A Multilevel Scheduling Model for Data Batches Processing in Conveyor Systems when Forming Sets and in the Presence of Restrictions.

Abstract. The paper considers the task of making schedules for processing data of various types in the conveyor system in the presence of time restrictions on its operation and under condition of forming sets from the processing results. The paper is dedicated to the first stage of solving the above problem. This stage is associated with the rationale of a multilevel model of making decisions on the composition of data batches; composition of groups of batches being processed within the specified time intervals of the system; scheduling of processing of data batches in groups taking into account the conditions of forming sets of various types from the processing results.

Keywords: multilevel programming model, data batches, groups of data batches, sets formed from processing results.

1. Введение. Конвейеризация программ предполагает, что за каждым сегментом конвейера закреплены для выполнения определенные их части [1]. Введем в рассмотрение следующие обозначения: i — идентификатор типа данных, обрабатываемых в системе, n — количество типов обрабатываемых данных ($i = \overline{1, n}$), n^i — количество данных i -го типа, которые должны быть обработаны. Данные i -го типа обрабатываются соответствующей им программой, тогда i — идентификатор программы, выполняемой в составе конвейера, обрабатывающей данные i -го типа. Однократное выполнение конвейеризированной программы i -го типа обеспечивает обработку одного элемента данных i -го типа. Если значение

$n^i \geq 1$ ($i = \overline{1, n}$), тогда обрабатывающая программа должна быть выполнена в конвейерной системе n^i раз. Программы, выполняющие обработку данных, находятся в оперативной памяти каждого из сегментов конвейера. Производительность всех сегментов конвейера является одинаковой, объемы вычислений, выполняемые в соответствии с программным кодом, реализующим обработку данных i -ых типов ($i = \overline{1, n}$) на каждом сегменте, различны. Тогда функционирование каждого сегмента конвейера при обработке данных характеризуется длительностями выполнения им операций с данными каждого i -го типа. Т. к. являются различными длительности выполнения программ на соответствующих сегментах, тогда может быть сформировано расписание обработки данных, представляющее собой порядок запуска программ на выполнение. В результате управление вычислительным процессом в конвейерных системах предполагает определение порядка запуска программ обработки данных на выполнение.

Постановка задачи предполагает задание значений $n^i > 1$ ($i = \overline{1, n}$). Однотипные данные могут быть объединены для обработки в партии. Партия — это совокупность данных одного типа, которые обрабатываются без переналадки (перенастройки) конвейера. Переналадка (перенастройка) сегмента конвейера представляет собой завершение активного состояния программы, выполнявшей обработку данных, и активизацию программы, которая будет реализовывать обработку данных в соответствии с сформированным расписанием. Характеристиками партии являются: тип i и количество данных в ней. Партия является фиксированной, если в нее входят все данные i -го типа. Если партия содержит не все n^i данных, в этом случае в системе обрабатывается ни одна, а несколько партий данных этого типа.

Постановка задачи также предполагает задание ограничений время функционирования системы при обработке данных. Т. е. задается количество интервалов времени функционирования системы при обработке данных и значения длительностей этих интервалов. Формируемые партии данных будут обработаны в течение этих интервалов времени. Совокупность партий, обрабатываемых в течение одного интервала времени функционирования системы, называется группой партий. Группы партий формируются в количестве, соответствующем задаваемому количеству интервалов времени функционирования системы.

Особенностью постановки задачи является наличие условия формирования комплектов разных типов из результатов обработки (составы комплектов разных типов являются заданными). Комплект данных (результатов обработки) — это совокупность результатов обработки разных типов, входящих в него в заданном количестве. То есть составы комплектов являются заданными и не могут быть изменены. Комплекты заданных составов формируются из данных, обрабатываемых в группах партий (обрабатываемых в течение заданных интервалов времени функционирования системы). При формировании комплектов предполагается, что данные всех n типов в определенном количестве входят в комплекты каждого типа.

В соответствии с выполненными рассуждениями управление вычислительным процессом в конвейерной системе предполагает: 1) определение составов партий данных, обрабатываемых в системе; 2) определение составов групп партий (совокупности партий), обрабатываемых в течение интервалов времени функционирования системы, количество и длительности которых задаются в качестве исходных данных решаемой задачи; 3) определение порядка обработки партий данных, входящих в каждую из групп партий (расписаний обработки партий данных). Для решения поставленной задачи определения составов партий, групп партий и порядков обработки партий применены методы, реализующие локальную оптимизацию решений (это связано с доказанным свойством NP-полноты рассматриваемой задачи). Поэтому эффективными решениями по составам партий, групп партий и порядков обработки партий будут являться те решения, которые при реализации локального поиска позволяют получить минимальные значения введенных критериев оптимизации (локально эффективные решения).

2. Анализ публикаций. Развитие современных методов построения расписаний обработки партий представлено в работах [2-11]. В [2] выполнена классификация задач управления обработкой партий (построения расписаний обработки партий). Задачи различаются по виду процесса обработки (непрерывный либо дискретный), способам представления времени моделирования (непрерывное либо дискретное), способам формирования партий и т.д. Управление обработкой партий предполагает построение расписаний для фиксированных партий, определение количества и размера партий до реализации процедуры построения расписаний (алгоритм определения составов партий никак не связан с характеристиками оборудования и процессом обработки, составы партий определяются без связи с построением расписаний),

определение размеров партий совместно с решением задачи построения расписаний их обработки. В [2] рассматривается задача формирования партий и распределения их по обрабатывающим приборам при дискретном времени моделирования для непрерывного производства. При ее решении использован аппарат частично целочисленного линейного программирования (ЧЦЛП). Большая размерность модели (наличие тридцати одного ограничения), использование методов ЧЦЛП, задание ограниченного количества обрабатывающих приборов (ограниченного количества ресурсов) делают затруднительным определение решения для задач большой размерности за ограниченное время. Решение задач обработки партий рассматривается в работе [3], однако указанные там методы построения расписаний предполагают наличие фиксированных партий данных при их обработке на ограниченное количество приборов (задача определения оптимальных составов партий в этой работе не рассматривается). В работах [4, 5] рассматривается задача управления обработкой партий в непрерывном (химическом) производстве. Под партиями в этом случае подразумевается объемы материалов, участвующих в процессе производства (объемы партий определяются способом производства требуемого вида продукта, количеством конечного продукта, определяемым в соответствии со спросом). Таким образом, в [4, 5] реализуется распределение обработки партий (размер которых не оптимизируется) материалов различных видов, обеспечивающих выпуск продуктов разных типов, по параллельно действующим машинам. При этом рассматриваются задачи ограниченной размерности (2 продукта, 2-3 прибора в обрабатывающей конвейерной системе). В работе [6] рассматривается решение задачи среднесрочного планирования выпуска продукции при ограниченном количестве ресурсов и последующего составления расписаний обработки сформированных при планировании партий. При этом под партией подразумевается совокупность изделий одного типа, выпуск которых закрепляется за определенным предприятием или производственным участком. Размеры партий определяются в соответствии с заказами на производство и директивными сроками их выпуска. Модель оптимизации составов партий учитывает только стоимостные параметры выпуска продукции, но не учитывает временные характеристики и особенности технологических процессов. На основе полученного решения по распределению заказов по производственным участкам (сформированным составам партий изделий) решается задача выделения для них ресурсов с целью обработки. Совместно задача планирования (определения составов

партий) и управления выпуском (формирования расписаний обработки партий) в работе не решается. В [6] использованы модель большой размерности и аппарат ЧЦП, что ограничивает размерность решаемой задачи. В работе [7] решается задача определения количества и составов партий единичных (разнотипных) требований, обрабатываемых на одном приборе, с заданными директивными сроками обработки и стоимостью доставки партий. Составы партий разнотипных требований формируются с учетом директивных сроков. Задача предполагает наличие одного обрабатывающего прибора, для которого отсутствуют простои при обработке партий. Тогда формирование партий предполагает и одновременное автоматическое построение расписания их обработки (т.е. вопрос оптимизации использования ресурсов обрабатывающих приборов в данной работе не рассматривается). В результате решается только задача определения составов партий на основе заданных директивных сроков окончания обработки требований. Аналогичная задача формирования партий требований в соответствии с их директивными сроками завершения, обрабатываемых на параллельных машинах, рассматривается в [8]. Объединение требований в партии реализуется с использованием эвристической процедуры в соответствии со значениями параметров начала обработки и директивными сроками окончания обработки требований. Так как обработка двух партий на одной машине (приборе) не может пересекаться, тогда сроки начала и окончания обработки партий используются для их распределения по параллельным машинам (расписание вытекает из сформированных составов партий). В работе [9] решается задача планирования производства полупроводников, предполагающая совместное формирование составов партий обрабатываемых пластин разных типов и расписаний обработки партий в конвейерной системе с параллельно действующими машинами. При определении локально оптимальных решений по составам партий и расписаниям их обработки используется один обобщенный критерий, в котором совместно учитываются эти решения. Особенностью аппарата формирования составов партий является использование метода отжига. Реализуется поиск в окрестностях текущего локально эффективного решения, для формирования новых решений, входящих в окрестности, введены эвристические правила, оперирующие со случайно выбираемыми партиями, изменяющие как составы партий, так и расписания их обработки (изменение позиции партии, изменение обрабатывающего прибора для выполнения операции, создание новых партий). На основе сформированного совместного решения по составам партий и

расписаний их обработки выполняется оценка эффективности полученного расписания с использованием дизъюнктивного графа. Предложенный в [9] подход позволяет реализовывать стохастический поиск локально оптимальных решений. Планированию производственного процесса посвящена работа [10], в которой решается задача распределения заказов на производство продукции по сменным заданиям, распределения заданий по партиям и формирование расписаний обработки партий. Для определения составов сменных заданий используется эвристическая процедура, для определения составов партий разработана имитационная процедура, позволяющая моделировать прохождение партиями конвейерной системы. Определение эффективных составов партий предполагает задание параметров имитационной модели, соответствующих размерам партий изделий каждого типа, и проведение моделирования. В соответствии с результатами моделирования выбираются значения параметров размеров партий, обеспечивающие минимальное значение критерия. После того как с использованием моделирования определены оптимальные размеры партий, выполняется формирование расписаний их обработки. Работа [11] реализует решение задачи определения составов партий компонент, из которых выполняется формирование элементов (аналог формирования комплектов из обработанных в системе изделий). Для формирования партий и расписаний их обработки введена оптимизационная модель, являющаяся многопараметрической и многоиндексной. Определение решений по количеству и составам партий осуществляется в работе путем полного перебора возможных значений этих параметров. Расписание для полученного решения формируется посредством использования эвристической процедуры. В итоге при большой размерности задачи (значительное число типов компонент и количество компонент каждого типа) прямой перебор при формировании партий не обеспечивает решение поставленной задачи за ограниченное время.

Таким образом, решение задачи определения составов партий и построения расписаний их обработки реализуется путем привлечения: 1) аппарата ЧЦЛП (однако при большой размерности задачи получение ее решения за ограниченное время является затруднительным, решения по составам партий формируются без учета расписаний их обработки); 2) методов формирования партий с учетом директивных сроков окончания обработки входящих в них требований (однако формирование групп партий, обрабатываемых в течение заданных интервалов времени, с использованием данных

методов является затруднительным); 3) эвристических процедур и правил (однако применение правил не позволяет получить решения, приближающиеся к оптимальным). В соответствии с этим разработка моделей и методов определения оптимальных составов партий, групп партий и расписаний их обработки является актуальной задачей.

3. Постановка цели и задач научного исследования. Цель работы состоит в совершенствовании методов построения расписаний обработки партий данных в конвейерных системах. Совершенствование методов связано с применением иерархического подхода к решению задачи определения расписаний обработки партий в течение задаваемых интервалов времени функционирования системы. Применение иерархического подхода предусматривает обоснование моделей и методов определения решений на каждом из уровней, на которых функционируют алгоритмы, реализующие формирование составов партий, составов групп партий, обрабатываемых в течение интервалов времени. В основу разрабатываемых методов положен подход к определению эффективных решений, связанный с их локальной оптимизацией (используется подход, связанный с реализацией поиска локально эффективных решений).

4. Основное содержание работы. Рассматриваемая задача является задачей с полной информацией, все параметры, характеризующие обрабатываемые данные (типы данных, количество данных, длительности обработки данных различных типов, составы комплектов различных типов и т.д.) и функционирующую систему (количество сегментов, дисциплина обработки партий, длительности интервалов времени функционирования системы и т.д.) являются заданными. Т.к. через i обозначен идентификатор типа данных, обрабатываемых в системе, тогда через d_i обозначим момент времени поступления в систему данных каждого i -го типа ($i = \overline{1, n}$). Для всех программ моменты времени их загрузки в систему и моменты поступления данных на обработку одинаковы, при этом $d_i = 0$. Обозначим через l индекс сегмента конвейера, осуществляющего выполнение l -й части программы, при этом $l = \overline{1, L}$. Каждым сегментом конвейера выполняются вычисления, соответствующие назначенной для него части программы. Дисциплина обслуживания выполняемых в системе программ предполагает прохождение данными, которые они обрабатывают, всех сегментов конвейера, при этом прерывание обработки данных является недопустимым. Выполнение на каждом l -м сегменте назначенной ему части i -го

программы характеризуется параметром длительности обработки данных, однозначно соответствующей объему вычислений.

Для классификации решаемой задачи построения расписаний выполнены следующие рассуждения. В соответствии с [12] задача составления расписаний может быть представлена в следующем виде: $\alpha|\beta|\gamma$, где параметр α соответствует классу (типу) обрабатываемой системы, β соответствует типу решаемой задачи и характеризует ее особенности, γ – вид критерия оптимизации для определения эффективного решения. Рассматривается конвейерная система, которая в [3] и [12] определена как *FlowShop*. Тогда в качестве α указывается F . Решаемая в системе задача обработки данных предполагает наличие ограничений на ресурсы, тогда в качестве характеристики β задачи указывается *res* [12]. Также решение задачи предполагает объединение обрабатываемых данных в партии. В этом случае в соответствии с [3] в качестве дополнительной характеристики решаемой задачи указывается *batch*. Критерий, в соответствии с которым определяется эффективное расписание, учитывает особенности метода определения порядка обработки партий, реализующего жадный подход. Метод предполагает, что в последовательности обработки партий, порядок выполнения операций с которыми оптимизирован на предыдущих шагах алгоритма, добавляется новая партия данных и определяется ее эффективное местоположение в этих последовательностях. Критерий эффективности решения по порядку обработки партий данных сформирован с точки зрения условия реализации внутренней цели функционирования системы, которая обуславливает необходимость эффективного использования оборудования системы (эффективное использование оборудования системы связано с минимизацией простоев сегментов конвейера при реализации обработки партий). Для классификации задачи в рассмотрение введено обозначение простоя (ожидания) отдельного l -го сегмента конвейера при обработке им партий в виде $T_l^{ож}$. Критерий, используемый при определении локально эффективного расписания обработки партий, учитывает суммарные простои всех сегментов конвейера при реализации операций. Для классификации задачи критерий обозначен в виде

$\sum_{l=1}^L T_l^{ож}$. Тогда рассматриваемая задача с точки зрения [3], [12] может

быть охарактеризована как: $F | res, batch | \sum_{l=1}^L T_l^{ож}$.

Особенностью решаемой задачи является необходимость формирования комплектов из результатов обработки данных. Комплект результатов обработки данных представляет собой совокупность данных, при этом количество данных каждого типа, входящих в комплект одного типа, является заданным. Через g обозначим идентификатор типа комплекта, который формируется из результатов обработки данных в конвейерной системе, а через $g^{ком}$ — общее количество типов формируемых комплектов. Через w_{ig} обозначим количество данных i -го типа, которые должны входить в один комплект g -го типа. В рассмотрение введена матрица (W) , значения элементов w_{ig} которой равны числу данных i -го типа, входящих в один комплект g -го типа.

Обработка партий данных разных типов выполняется с учетом ограничений на время функционирования системы, тогда для идентификации интервалов времени в рассмотрение введены обозначения: t^z — интервал времени, в течение которого может быть реализована обработка партий данных; Z — количество этих интервалов ($z = \overline{1, Z}$). Обработка партий данных при наличии ограничений (интервалов t^z , $z = \overline{1, Z}$) должна быть реализована с точки зрения внешней цели функционирования системы, для достижения которой количество сформированных комплектов должно быть maximизировано. То есть обработка данных направлена на осуществление операций на всех сегментах конвейера с данными разных типов таким образом, чтобы из результатов обработки могло быть сформировано максимальное количество комплектов. Формирование партий данных i -ых типов ($i = \overline{1, n}$) вместо фиксированных партий позволяет получить более эффективное решение (с точки зрения внешней цели функционирования системы — максимизации количества комплектов). При этом должно быть определено как количество партий данных, так и их составы. Так как обработка партий данных i -ых типов ($i = \overline{1, n}$) реализуется в течение заданных интервалов времени функционирования системы

t^z ($z = \overline{1, Z}$), тогда на основе решения по количеству и составам партий формируются группы партий, каждая из которых обрабатывается в течение одного интервала t^z ($z = \overline{1, Z}$).

Входными данными для системы построения расписаний обработки партий являются: типы и количество данных, обрабатываемых в системе; значения интервалов времени t^z функционирования системы при обработке данных; количество Z интервалов времени; матрица (W) , элемент w_{ig} которой равен количеству данных i -го типа, которые должны входить в один формируемый комплект g -ого типа ($i = \overline{1, n}$; $g = \overline{1, g^{ком}}$). Выходными решениями являются: количество и составы партий данных i -ых типов ($i = \overline{1, n}$); составы групп партий, обрабатываемых системой в течение заданных интервалов времени t^z , определяемые в соответствии с условием формирования из обрабатываемых в группах данных максимального количества комплектов; расписания обработки партий данных каждой группы. В итоге обобщенная цель функционирования системы представлена как совокупность иерархически упорядоченных целей подсистем (функций, решаемых на иерархически упорядоченных уровнях системы построения расписаний обработки партий). Формирование решений на уровнях системы построения расписаний обработки партий осуществляется следующим образом: первый уровень — решения по количеству и составам партий, второй уровень — решения по составам групп партий; третий уровень — решения по порядку обработки входящих в группы партий на сегментах конвейера. Выполненная декомпозиция обобщенной функции системы на совокупность иерархически упорядоченных подфункций позволяет рассмотреть задачу как задачу теории иерархических игр [13, 14].

С целью формирования модели иерархической игры для построения расписаний обработки партий в рассмотрение введены следующие обозначения: m_i — количество партий данных i -го типа ($i = \overline{1, n}$), формируемых на первом уровне принятия решений, при $i = \overline{1, n}$ элементы m_i образуют вектор (M) ; (A) — матрица, элемент a_{ih} которой — это количество данных i -го типа в h -ой партии ($h \leq m_i$), размерность матрицы (A) — $n \times h^{max}$, где $h^{max} = \max_{1 \leq i \leq n} (m_i)$. Решение, формируемое на первом уровне системы (количество и составы партии)

имеет вид: $[(M), (A)]$. Через N^z обозначим группу партий, обрабатываемых в течение одного из интервалов t^z ($z = \overline{1, Z}$). Партии данных i -го типа могут входить в различные группы партий N^z ($z = \overline{1, Z}$). Через m_i^z обозначим количество партий данных i -го типа в группе партий N^z , через $(A)_i^z$ — вектор количества данных i -го типа в m_i^z партиях в группе N^z . Партии данных i -го типа, входящие в группу N^z , определены с использованием набора: $[i, m_i^z, (A)_i^z]$. Группа N^z — совокупность наборов: $N^z = \{ [i, m_i^z, (A)_i^z]_k \mid k = \overline{1, k_z} \}^z$, где k_z — количество типов данных, партии которых входят в N^z . При $m_i^z = 0$ набор параметров $[i, m_i^z, (A)_i^z]$ не входит в группу партий N^z . Решение, формируемое на втором уровне системы — совокупность групп партий, имеет вид: $\{ N^z \mid (z = \overline{1, Z}) \}$.

Расписание обработки партий группы N^z обозначено как π^z , оно представляет собой совокупность последовательностей π^l запуска партий на обработку на каждом l -ом сегменте конвейера ($l = \overline{1, L}$). Расписание π^z имеет вид: $\pi^z = \{ \pi^1, \pi^2, \pi^3, \dots, \pi^L \}^z$. Расписание обработки партий π^z формируется в предположении, что порядок обработки партий является одинаковым на всех L сегментах конвейера. Для формализации вида последовательностей π^l расписания π^z в рассмотрении введена матрица порядка обработки партий в системе $(P)^z$; элемент $p_{ij}^z = 1$, если партия данных i -го типа занимает в последовательности π^l j -ю позицию, $p_{ij}^z = 0$ в противном случае, размерность матрицы $k_z \times n_p^z$, где k_z — количество типов данных в партиях группы N^z , n_p^z — количество партий в последовательностях π^l расписания π^z . Порядок обработки партий группы на всех сегментах одинаков, поэтому достаточно определения одной матрицы порядка $(P)^z$. В рассмотрение введена матрица $(R)^z$ — матрица количества данных i -ых типов в партиях, занимающих в последовательности π^l j -е позиции (элемент r_{ij}^z равен

количеству данных i -го типа в партии, занимающей j -ю позицию в π^l , размер матрицы $k_z \times n_p^z$). Решение, формируемое на нижнем уровне иерархии, имеет вид: $\{f(P)^z, (R)^z | z = \overline{1, Z}\}$.

С точки зрения аппарата теории иерархических игр каждым игроком принимается решение, соответствующее функции (подцели) отдельного уровня. Игрок на верхнем уровне принимает решение по составам партий данных, игрок на втором уровне принимает решение по составам групп партий, игрок на третьем уровне принимает решение по расписанию обработки партий данных в конвейерной системе.

В соответствии с функциями уровней системы между ними выполняется обмен информацией следующего вида: 1) на вход первого уровня подаются: количество n типов обрабатываемых данных; количество n^i ($i = \overline{1, n}$) данных каждого типа, обрабатываемых в системе; матрица (W) составов комплектов; с выхода уровня передаются составы партий данных i -ых типов ($i = \overline{1, n}$) — решение $[f(M), (A)]$; 2) на вход второго уровня — составы партий данных; с выхода уровня — сформированные с учетом интервалов времени t^z группы партий данных N^z (решение $\{N^z | (z = \overline{1, Z})\}$); 3) на вход третьего уровня — решение $\{N^z | (z = \overline{1, Z})\}$; с выхода уровня — сформированные расписания π^z ($z = \overline{1, Z}$). При распределении совокупности партий данных i -ых типов (представленной в решении $[f(M), (A)]$) по группам партий N^z ($z = \overline{1, Z}$) состав партий не изменяется (значения m_i и a_{ih} , поступившие с верхнего уровня, изменены быть не могут). В общем виде модель иерархической игры в соответствии с введенными обозначениями имеет следующий вид [14, 15]:

1) первый уровень:

$$f_1([M, A], \{N^{z*} | z = \overline{1, Z}\}) \rightarrow \min(\max); \quad (1)$$

2) второй уровень:

$$f_2(\{N^z | z = \overline{1, Z}\}, \{[(P)^z, (R)^z]^* | z = \overline{1, Z}\}) \rightarrow \min(\max); \quad (2)$$

3) третий уровень:

$$f_3(\{N^z | z = \overline{1, Z}\}, \{[(P)^z, (R)^z] | z = \overline{1, Z}\}) \rightarrow \min(\max). \quad (3)$$

Комплекты формируются только из данных, входящих в партии, включенные в соответствующие группы (партий, которые будут обработаны в соответствии с построенными расписаниями). Поэтому для текущих составов партии наилучшим решением по составам групп партий будет являться решение, которое обеспечивает максимальное количество формируемых комплектов. Данное условие соответствует внешней цели функционирования системы, в соответствии с которой необходимо реализовать выпуск большего количества комплектов в течение заданных временных интервалов. Тогда для определения эффективности решений на втором уровне в качестве критерия используется количество комплектов, сформированных из данных, обработанных в составе групп партий. Решение $\{ N^{z*} | (z = \overline{I, Z}) \}$ соответствует максимальному количеству комплектов разных типов, которые могут быть сформированы из данных, обработанных в группах.

Количество данных, обработанных в течение интервалов t^z , зависит от количества и составов сформированных партий (составы групп партий зависят от количества и составов партий). Решение по порядку обработки партий в каждой из групп является зависящим от составов партий данных в этой группе. Таким образом, определение количества и составов партий данных различных типов, распределение партий по группам, построение расписаний обработки партий групп с учетом ограничений на длительность интервалов t^z должно обеспечить формирование максимального количества комплектов.

Для формирования решения по составам комплектов данных на основе решения по группам $\{ N^z | (z = \overline{I, Z}) \}$ в рассмотрение введена матрица (N^{KOM}) , значения элементов g -го столбца которой соответствуют количествам данных i -ых типов, включаемых в комплекты g -го типа (элемент n_{ig}^{KOM} соответствует количеству данных i -го типа, которые будут входить в состав комплектов g -го типа). Для идентификации количества комплектов, сформированных из данных, обработанных в группах $N^z (z = \overline{I, Z})$, в рассмотрение введен вектор N^K , значение g -го элемента которого соответствует количеству комплектов g -го типа (элемент n_g^K — количество комплектов g -го типа, сформированных из данных, обработанных в группах $N^z (z = \overline{I, Z})$). Для реализации алгоритма определения составов комплектов введен вектор N^{gz} , элемент n_i^{gz} — это количество данных

i -го типа, обрабатываемых в группах партий $N^z (z = \overline{1, Z})$, которые могут быть распределены по комплектам g^{KOM} типов. Перед началом реализации алгоритма распределения данных по комплектам элемент

$$n_i^{gz} \text{ определяется следующим образом: } n_i^{gz} = \sum_{z=1}^Z \sum_{h=1}^{m_i^z} (a_h)_i^z.$$

Алгоритм формирования комплектов предполагает, что за одну итерацию данные каждого типа в количестве $n_i^{gz} (i = \overline{1, n})$ равномерно распределяются по комплектам всех g^{KOM} типов. При формировании комплекта некоторого g -го типа для всех n типов данных выполняется проверка условий $n_i^{gz} \geq w_{ig} (i = \overline{1, n})$. Если для каждого i -го типа данных ($i = \overline{1, n}$) введенное условие выполняется, тогда в комплект рассматриваемого g -го типа выделяется по w_{ig} данных каждого типа. В этом случае количество комплектов рассматриваемого g -го типа увеличивается: $n_g^K = n_g^K + 1$, выполняется изменение значений n_i^{gz} для каждого i -го типа данных ($i = \overline{1, n}$): $n_i^{gz} = n_i^{gz} - w_{ig}$, количество данных i -ых типов, использованных при формировании комплекта g -го типа изменяется: $n_{ig}^{KOM} = n_{ig}^{KOM} + w_{ig} (i = \overline{1, n})$. На следующем шаге реализуется формирование одного экземпляра комплекта $(g+1)$ -го типа и т.д. Итерации алгоритма продолжаются до тех пор, пока хотя бы для одного из типов комплектов и всех n типов данных выполняется условие $n_i^{gz} \geq w_{ig} (i = \overline{1, n})$. После распределения данных i -ых типов в количестве $n_i^{gz} (i = \overline{1, n})$ по комплектам количество

комплектов различных типов вычисляется выражением $\sum_{g=1}^{g^{KOM}} n_g^K$.

Количество данных i -го типа, вошедших во все сформированные комплекты определяется следующим образом: $\sum_{g=1}^{g^{KOM}} n_g^K \cdot w_{ig}$. Количество

данных i -го типа, обрабатываемых в группах партий $N^z (z = \overline{1, Z})$, но не включаемых в составы ни одного из сформированных комплектов,

определяется выражением: $\sum_{z=1}^Z \sum_{h=1}^{m_i^z} (a_h)_i^z - \sum_{g=1}^{g^{KOM}} n_g^k \cdot w_{ig}$. Общее количество данных различных типов, обрабатываемых в группах партий, но не включаемых в комплекты, вычисляется выражением:

$$\sum_{i=1}^n \left(\sum_{z=1}^Z \sum_{h=1}^{m_i^z} (a_h)_i^z - \sum_{g=1}^{g^{KOM}} n_g^k \cdot w_{ig} \right).$$

Для определения вида модели вычислительного процесса обработки партий данных, включенных в группы N^z ($z = \overline{1, Z}$) (при наличии ограничений на длительность интервалов t^z), введены обозначения: t_i^l — длительность обработки данных i -го типа на l -ом сегменте конвейера ($l = \overline{1, L}$); t_{ik}^l — длительность переналадки l -го сегмента с обработки данных i -го типа на обработку данных k -го типа; t_{ii}^l — длительность первоначальной наладки l -го сегмента на обработку данных i -го типа; t_{ji}^{nl} — момент времени начала обработки партии данных i -го типа, занимающей в последовательности π^l j -ю позицию; $(t_{ji}^{nl})^z$ — матрица моментов времени начала обработки партий данных i -ых типов, занимающих в π^l j -е позиции (для группы партий N^z); t_{jq}^{ol} — момент времени начала обработки данных с q -ым порядковым номером в партии, занимающей j -ю позицию в последовательностях π^l ; $(t_{jq}^{ol})^z$ — матрица моментов времени начала обработки q -ых данных в партии, занимающей в π^l j -ю позицию ($q = \overline{1, n_j^z}, n_j^z = \sum_{h=1}^{n^z} r_{hj}^z$, где n_j^z — количество данных в партии, входящей в группу N^z , занимающей j -ю позицию в π^l). С использованием элементов матриц $(P)^z$ и $(t_{jq}^{ol})^z$ элементы матрицы $(t_{ji}^{nl})^z$ определяются следующим образом: $[t_{ji}^{nl}]^z = p_{ij}^z \cdot [t_{jl}^{ol}]^z$, где $i = \overline{1, k_z}, j = \overline{1, n_p^z}$, $[t_{jl}^{ol}]^z$ — момент времени начала обработки первых данных в партии, занимающей j -ю позицию в π^l . В рассмотрение введена матрица переналадок (t_{ik}^l) ,

элементы t_{ik}^l которой соответствуют длительностям переналадки сегментов с обработки данных i -го типа на обработку данных k -го типа, элементы t_{ii}^l — длительности первоначальной наладки сегментов на обработку данных i -ых типов. Выполним дальнейшие рассуждения по обоснованию модели вычислительного процесса обработки партий данных в конвейерной системе для одной из групп N^z (опуская при этом индекс группы z).

Формализация выражений для определения параметров модели вычислительного процесса t_{ji}^{nl} , t_{qj}^{ol} ($i = \overline{1, k_z}, j = \overline{1, n_p}^l, l = \overline{1, L}$) выполнена следующим образом. Для первого сегмента выражения для t_{jq}^{ol} формируются следующим образом: 1) если t_{11}^{ol} — момент времени начала обработки данных ($q=1$) в партии, занимающей первую позицию в последовательности π^l , t_{1i}^{nl} — момент времени начала обработки партии данных i -го типа, занимающей в π^l первую позицию, тогда $t_{11}^{ol} = t_{1i}^{nl} = t_{ii}^l$, 2) выражение $\sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}$ определяет интервал первоначальной наладки первого сегмента на обработку данных i -го типа; $(q-1) \sum_{h=1}^{k_z} t_h^l \cdot p_{hl}$ — время обработки предыдущих $(q-1)$ -их данных в этой партии, $(t)^l$ — вектор длительностей обработки на первом приборе данных (t_i^l — элемент вектора), тогда значения t_{1q}^{ol} моментов времени начала обработки любых данных в первой партии в π^l определяется выражением вида:

$$t_{1q}^{ol} = \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl} + (q-1) \sum_{h=1}^{k_z} t_h^l \cdot p_{hl}; \quad (4)$$

3) если t_{i,i_2}^{lnep} — интервал переналадки первого сегмента конвейера с обработки данных i -го типа (первая позиция партии в π^l) на обработку данных другого типа (вторая позиция партии в π^l), t_{2i}^{ol} — момент времени начала обработки данных в первой позиции во второй партии (t_{2i}^{nl} — начало обработки этой партии данных i -го типа в π^l

$(t_{2i}^{n1} = t_{2i}^{01})$), тогда с учетом (4) значения t_{2i}^{01} (t_{2i}^{n1}) определяются выражением вида:

$$t_{2i}^{n1} = t_{2i}^{01} = \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep}, \quad (5)$$

где значение $t_{i_1 i_2}^{Inep}$ определяется следующим образом:

$$t_{i_1 i_2}^{Inep} = t_{i_r i_r'}^l, \text{ где } \begin{cases} i_r = i \mid p_{i_l}^l = l, i = \overline{1, k_z} \\ i_r' = j \mid p_{j_2}^l = l, j = \overline{1, k_z} \end{cases}.$$

Выражение $\sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep}$ позволяет определить момент времени начала обработки второй в π^l партии данных, выражение $(q-1) \sum_{h=1}^{k_z} t_h^l \cdot p_{h2}^l$ — длительность обработки данных, предшествующих данным в q -ой позиции в этой партии, тогда время начала обработки q -ых данных i -го типа в партии, занимающей вторую позицию в π^l , определяется следующим образом:

$$t_{2q}^{01} = \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep} + (q-1) \sum_{h=1}^{k_z} t_h^l \cdot p_{h2}^l, \quad q = \overline{1, n_2^z}. \quad (6)$$

По аналогии с (5), (6) сформированы выражения для вычисления моментов времени начала обработки третьей, четвертой партий и данных в q -ых позициях в них:

$$\begin{aligned} t_{3i}^{n1} &= \sum_{p=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep} + \sum_{h=1}^{k_z} t_h^l \cdot r_{h2}^l + t_{i_2 i_3}^{Inep}; \\ t_{3q}^{01} &= \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep} + \sum_{h=1}^{k_z} t_h^l \cdot r_{h2}^l + t_{i_2 i_3}^{Inep} + (q-1) \sum_{h=1}^{k_z} t_h^l \cdot p_{h3}^l; \\ t_{4i}^{n1} &= \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep} + \sum_{h=1}^{k_z} t_h^l \cdot r_{h2}^l + t_{i_2 i_3}^{Inep} + \sum_{h=1}^{k_z} t_h^l \cdot r_{h3}^l + t_{i_3 i_4}^{Inep}; \\ t_{4q}^{01} &= \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{h=1}^{k_z} t_h^l \cdot r_{hl}^l + t_{i_1 i_2}^{Inep} + \sum_{h=1}^{k_z} t_h^l \cdot r_{h2}^l + t_{i_2 i_3}^{Inep} + \\ &+ \sum_{h=1}^{k_z} t_h^l \cdot r_{h3}^l + t_{i_3 i_4}^{Inep} + (q-1) \sum_{h=1}^{k_z} t_h^l \cdot p_{h4}^l. \end{aligned}$$

Тогда моменты времени начала обработки партии данных i -го типа, занимающей j -ю позицию в π^l , моменты времени начала обработки q -ых данных в партии в j -й позиции в π^l , определяются выражениями вида:

$$t_{ji}^n = \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{f=1}^{j-1} \sum_{h=1}^{k_z} t_h^l \cdot r_{hf}^l + \sum_{h=1}^{j-1} t_{i_h t_{h+1}}^{l nep},$$

$$t_{jq}^{0l} = \sum_{h=1}^{k_z} t_{hh}^l \cdot p_{hl}^l + \sum_{f=1}^{j-1} \sum_{h=1}^{k_z} t_h^l \cdot r_{hf}^l + \sum_{h=1}^{j-1} t_{i_h t_{h+1}}^{l nep} + (q-l) \sum_{h=1}^{k_z} t_h^l \cdot p_{hj}^l.$$

Выражение для t_{jq}^{0l} ($q = \overline{1, n_j^z}$, при $l > 1$) сформированы следующим образом. Для $l=2$ и $j=1$ имеем:

$$t_{11}^{02} = t_{1i}^{n2} = \max\left(\sum_{h=1}^{k_z} t_{hh}^2 \cdot p_{h1}; t_{11}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h1}\right);$$

$$t_{12}^{02} = \max\left(t_{11}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h1}; t_{12}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h1}\right);$$

$$t_{1n_1}^{02} = \max\left(t_{1, n_1-1}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h1}; t_{1n_1}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h1}\right), \text{ где } n_1^z = \sum_{i=1}^{k_z} r_{i1}^z.$$

Выражения для определения t_{2q}^{02} ($q = \overline{1, n_2^z}$, $n_2^z = \sum_{h=1}^{k_z} r_{h2}^z$) имеют вид:

$$t_{21}^{02} = t_{2i}^{n2} = \max\left(t_{1n_1}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h1} + t_{i_1 i_2}^{2 nep}; t_{21}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h,2}\right);$$

$$t_{22}^{02} = \max\left(t_{21}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h2}; t_{22}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h2}\right);$$

$$t_{2n_2}^{02} = \max\left(t_{2, n_2-1}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h2}; t_{2, n_2}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h2}\right).$$

Вид выражений для t_{3i}^{02} и t_{3q}^{02} :

$$t_{31}^{02} = t_{3i}^{n2} = \max\left(t_{2n_2}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h2} + t_{i_2 i_3}^{2 nep}; t_{31}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h3}\right);$$

$$t_{3q}^{02} = \max\left(t_{3, q-1}^{02} + \sum_{h=1}^{k_z} t_h^2 \cdot p_{h3}; t_{3q}^{01} + \sum_{h=1}^{k_z} t_h^1 \cdot p_{h3}\right), \text{ где } q = \overline{2, n_3^z}, n_3^z = \sum_{h=1}^{k_z} r_{h3}^z.$$

Сформированы обобщенные выражения для t_{jq}^{0l} (с учетом особенностей для t_{ll}^{0l} и t_{jl}^{0l} , $l = \overline{2, L}$, $q = \overline{1, n_j^z}$, $n_j^z = \sum_{h=1}^{k_z} r_{hj}^z$) в виде:

$$t_{ll}^{0l} = t_{li}^{nl} = \max\left(\sum_{h=1}^{k_z} t_h^l \cdot p_{hl}; t_{ll}^{0l-1} + \sum_{h=1}^{k_z} t_h^{l-1} \cdot p_{hl}\right);$$

$$t_{jl}^{0l} = t_{ji}^{nl} = \max\left(t_{j-1, n_{j-1}}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{h, j-1} + t_{i_{j-1}^l}^{lnep}; t_{jl}^{0l-1} + \sum_{h=1}^{k_z} t_h^{l-1} \cdot p_{hj}\right);$$

$$t_{jq}^{0l} = \max\left(t_{jq-1}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{hj}; t_{jq}^{0l-1} + \sum_{h=1}^{k_z} t_h^{l-1} \cdot p_{hj}\right).$$

Полученные выражения представляют собой модель вычислительного процесса обработки партий данных i -ых типов ($i = \overline{1, n}$) на l -ых сегментах конвейера ($l = \overline{1, L}$). Метод построения расписаний обработки партий данных на третьем уровне системы реализует жадный подход к оптимизации. Он предполагает добавление текущей рассматриваемой партии в конец последовательностей π^l ($l = \overline{1, L}$), в которых на предыдущих шагах алгоритма были размещены партии данных, и определение эффективного местоположения этой партии в π^l (определение для рассматриваемой партии позиций в π^l ($l = \overline{1, L}$), которые обеспечивают локально оптимальное решение). Так как в задаче заданы ограничения на интервалы времени функционирования системы, учитываемым ресурсом является время работы системы. В этом случае решение по порядку обработки партий должно быть построено таким образом, чтобы временной ресурс системы использовался в полной мере. Тогда критерий эффективности расписаний обработки партий соответствует внутренней цели функционирования системы, определяющей необходимость минимизации простоев оборудования при обработке. При определении текущей эффективной позиции рассматриваемой партии в π^l критерий характеризует простои сегментов конвейера при обработке текущего количества партий, находящихся в π^l .

В этом случае критерий эффективности расписания обработки партий на нижнем уровне учитывает: а) время простоя сегментов в

ожидании начала обработки партий данных (с учетом интервалов наладки, переналадки и последующего ожидания); б) время простоя сегментов в ожидании готовности данных при их обработке внутри партий. Простой l -го сегмента в ожидании обработки первой в π^l ($l = \overline{1, L}$) партии равен значению t_{1l}^{0l} , суммарное время простоя сегментов в ожидании начала обработки партий в последовательностях π^l определяется выражением $\sum_{l=1}^L t_{1l}^{0l}$. Интервал

простоя l -го сегмента в ожидании начала обработки следующей партии после окончания обработки предыдущей определяется выражением вида: $t_{j1}^{0l} - \left[t_{j-1, n_{j-1}^z}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{h, j-1} \right]$, где $j > 1$,

$n_{j-1}^z = \sum_{h=1}^{k_z} r_{h, j-1}^z$ — количество данных в предшествующей $(j-1)$ -ой партии. Суммарный простой l -го сегмента в ожидании начала обработки всех j -ых партий ($j = \overline{1, n_p^z}$) определен следующим образом:

$$\sum_{j=2}^{n_p^z} \left[t_{j1}^{0l} - \left[t_{j-1, n_{j-1}^z}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{h, j-1} \right] \right].$$

В этом случае суммарной простоя всех L сегментов в ожидании начала обработки партий на них (простой между партиями для всех L сегментов) определяется выражением:

$$\sum_{l=1}^L \sum_{j=2}^{n_p^z} \left[t_{j1}^{0l} - \left[t_{j-1, n_{j-1}^z}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{h, j-1} \right] \right]. \quad (7)$$

Простой l -го сегмента в ожидании готовности к обработке данных, занимающих q -ю позицию в j -ой партии в последовательности π^l , определяется выражением вида: $t_{jq}^{0l} - \left[t_{j, q-1}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{hj} \right]$, где $j = \overline{1, n_p^z}$. Это выражение соответствует интервалу между данными в q -ой и $(q-1)$ -ой позициях в j -ой партии в π^l . Тогда суммарный простой

l -го прибора в ожидании готовности к обработке всех данных в j -ой партии в π^l вычисляется с использованием выражения:

$$\sum_{q=2}^{n_j^z} \left[t_{jq}^{0l} - \left[t_{j,q-1}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{hj} \right] \right], \quad (8)$$

где q – номер позиции данных в j -ой партии в π^l , n_j^z — число данных в этой j -ой партии, $n_j^z = \sum_{h=1}^{k_z} r_{hj}$. На основе (8) общий простой l -го сегмента в ожидании готовности к обработке данных внутри всех партий в π^l определяется выражением вида:

$$\sum_{j=1}^{n_p^z} \sum_{q=2}^{n_j^z} \left[t_{jq}^{0l} - \left[t_{j,q-1}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{hj} \right] \right].$$

Суммарный простой всех L сегментов конвейера в ожидании готовности данных внутри партий вычисляется выражением:

$$\sum_{l=2}^L \sum_{j=1}^{n_p^z} \sum_{q=2}^{n_j^z} \left[t_{jq}^{0l} - \left[t_{j,q-1}^{0l} + \sum_{h=1}^{k_z} t_h^l \cdot p_{hj} \right] \right]. \quad (9)$$

При оценивании эффективности решения на нижнем уровне иерархии для текущего количества партий, добавленных в π^l ($l = \overline{1, L}$), учитывается общий простой сегментов конвейера при определении порядка обработки этих партий, и должны быть учтены: выражение $\sum_{l=1}^L t_{l1}^{0l}$ и формулы (7) и (9).

Рассматриваемая задача предполагает ограничение времени функционирования системы, тогда при формировании комплектов могут быть использованы только данные, партии которых будут обработаны в группах N^z ($z = \overline{1, Z}$). Максимизация количества комплектов, сформированных из данных, обработанных в составе групп партий, отвечает внешней цели функционирования системы. Тогда для достижения этой цели решение по составам групп $\{N^z | z = \overline{1, Z}\}$ должно анализироваться с точки зрения количества комплектов, которые могут быть сформировано из данных, входящих

в партии, включенные в группы N^z ($z = \overline{I, Z}$). Таким образом, на втором уровне иерархии системы на основе решения по составам групп партий $\{N^z | z = \overline{I, Z}\}$ необходимо определить решение по составам комплектов данных, формируемых из результатов их обработки в группах партий. Общее количество комплектов, сформированных из данных, партии которых включены в группы, определяется выражением, введенным ранее: $\sum_{g=1}^{g^{ком}} n_g^k$. Приведенное

выражение использовано в качестве критерия эффективности решений $\{N^z | z = \overline{I, Z}\}$ по составам групп партий.

На первом уровне иерархии принимается решение по составам партий данных в виде $[(M), (A)]$. В соответствии с видом модели (1) критерий эффективности решений на первом уровне должен учитывать как решение $[(M), (A)]$, формируемое на этом уровне, так и эффективное решение по составам групп партий, формируемое на втором уровне в виде $\{N^{z*} | z = \overline{I, Z}\}$. Решение $\{N^{z*} | z = \overline{I, Z}\}$ характеризуется количеством сформированных комплектов разных типов — вектором N^k (решение $\{N^{z*} | z = \overline{I, Z}\}$ обеспечивает максимальное количество комплектов). Тогда при определении значения критерия f_1 , характеризующего решение $[(M), (A)]$ на первом уровне, используется само это решение и решение по количеству комплектов каждого типа — вектор N^k , характеризующий эффективное решение по составам групп партий $\{N^{z*} | z = \overline{I, Z}\}$.

Входными параметрами для решаемой задачи являются: количество n типов данных, обрабатываемых в системе, количество данных каждого типа n^i ($i = \overline{1, n}$), длительности интервалов времени t^z ($z = \overline{I, Z}$) функционирования системы при обработке данных. Предполагается, что все данные, подаваемые на вход системы для обработки, должны быть использованы при формировании комплектов, то есть все данные, поданные на вход, обрабатываются системой и из них формируются комплекты. Тогда внешняя цель функционирования системы предполагает, что все данные, поданные на вход конвейера, обрабатываются и используются при формировании комплектов. В этом случае решение $[(M), (A)]$ охарактеризовано количеством данных, которые должны быть

обработаны в соответствии с задаваемыми входными характеристиками и включены в комплекты, но в соответствии с решением N^k в комплекты не вошли. В соответствии с решением $[(M), (A)]$ количество данных i -го типа, которое должно быть

обработано в системе определяется выражением вида: $\sum_{h=1}^{m_i} a_{ih}$, где

a_{ih} — элемент матрицы (A) (количество данных i -го типа в h -ой партии ($h = \overline{1, m_i}$)), m_i — количество партий данных i -го типа. Общее количество данных различных типов, которые должны быть

обработаны в системе, определяется выражением $\sum_{i=1}^n \sum_{h=1}^{m_i} a_{ih}$. Количество

данных различных типов, используемых при формировании комплектов, определяется выражением $\sum_{i=1}^n \sum_{g=1}^{g^{KOM}} n_{ig}^{KOM} \cdot w_{ig}$. Тогда

количество данных, которые должны быть обработаны в соответствии с решением $[(M), (A)]$, но не использованы при формировании комплектов, определяется выражением вида:

$$\sum_{i=1}^n \sum_{h=1}^{m_i} a_{ih} - \sum_{i=1}^n \sum_{g=1}^{g^{KOM}} n_{ig}^{KOM} \cdot w_{ig}.$$

Полученное выражение позволяет охарактеризовать достижение внешней цели функционирования системы при формировании составов партий данных.

Для задания индекса последней партии в последовательности π^L использовано обозначение n_p^z (где n_p^z — количество партий, входящих в группу N^z). Если через n_j^z обозначено количество данных в партии, входящей в группу N^z и занимающей j -ю позицию в последовательности π^L , тогда при $j = n_p^z$ через $(t_{j, n_j^z}^{0L})^z$ обозначен момент времени начала обработки последних данных в партии с индексом n_p^z на L -ом сегменте конвейера, а окончание обработки этой

партии определяется выражением $(t_{j, n_j^z}^{0L})^z + \sum_{i=1}^{k_z} (t_i^L)^z \cdot p_{i,j}^z$ (при $j = n_p^z$).

При задании значений t^z ($z = \overline{1, Z}$) ограничение на время обработки партий данных, входящих в группы N^z , имеет вид:

$[(t_{j,n_j}^{0L})^z + \sum_{i=1}^{k_z} (t_i^L)^z \cdot p_{i,j}^z] \leq t^z$ при условии $j = n_p^z$ ($z = \overline{1, Z}$). В

соответствии с выполненными рассуждениями модель многоуровневого программирования для определения эффективных составов партий, составов групп партий и расписаний обработки партий имеет вид (с учетом (1-3)):

$$1) \text{ первый уровень: } \min f_1, f_1 = \sum_{i=1}^n \sum_{h=1}^{m_i} a_{ih} - \sum_{i=1}^n \sum_{g=1}^{g^{KOM}} n_{ig}^{KOM} \cdot w_{ig};$$

$$2) \text{ второй уровень: } \max f_2, f_2 = \sum_{g=1}^{g^{KOM}} n_g^k;$$

$$3) \text{ третий уровень: } \min f_3^z \text{ (} z = \overline{1, Z} \text{), где:}$$

$$f_3^z = \sum_{l=2}^L [t_{l1}^{0l}]^z + \sum_{l=1}^L \sum_{j=2}^{n_p^z} \left[[t_{jl}^{0l}]^z - \left[[t_{j-1, n_{j-1}}^{0l}]^z + \sum_{h=1}^{k_z} [t_h^l]^z \cdot p_{h, j-1}^z \right] \right] + \\ + \sum_{l=2}^L \sum_{j=1}^{n_p^z} \sum_{q=2}^{n_j^z} \left[[t_{jq}^{0l}]^z - \left[[t_{j, q-1}^{0l}]^z + \sum_{h=1}^{k_z} [t_h^l]^z \cdot p_{hj}^z \right] \right].$$

4) ограничения на длительность реализации обработки партий в N^z

$$(z = \overline{1, Z}): \left[(t_{j,n_j}^{0L})^z + \sum_{i=1}^{k_z} (t_i^L)^z \cdot p_{i,n_j}^z \right] \leq t^z \text{ при } j = n_p^z \text{ (} z = \overline{1, Z} \text{)}.$$

Таким образом, модель многоуровневого программирования представлена в виде совокупности критериев оптимизации на каждом уровне иерархии принятия решений и ограничения на время функционирования системы при обработке данных.

5. Заключение. Результатом выполненных исследований является сформированная многоуровневая модель построения комплексных расписаний обработки партий с формированием из обработанных данных комплектов различных типов (при ограничении на время реализации выполнения операций с партиями в группах). Использование сформированной модели позволяет определить: 1) эффективные составы партий (количество партий, число данных в них), для которых общее количество данных, не вошедших в состав комплектов разных типов, будет минимальным; 2) эффективных составов групп партий, обрабатываемых в течение установленных

интервалов t^z , для которых общее количество сформированных комплектов будет максимальным; 3) эффективных порядков обработки партий на сегментах конвейера. Дальнейшие исследования направлены на разработку методов построения комплексных расписаний обработки партий данных при формировании комплектов и наличии ограничений на длительности интервалов реализации операций с партиями в группах, соответствующих сформированной модели.

Литература

1. *Хьюз К., Хьюз Т.* Параллельное и распределенное программирование на C++ // М.: Вильямс. 2004. 672 с.
2. *Mendez C.A. et al.* State-of-the-art review of optimization methods for short-term scheduling of batch processes // *Computers and Chemical Engineering*. 2006. vol. 30. pp. 913–946.
3. *Ковалев М.М.* Модели и методы календарного планирования: Курс лекций // Минск: Изд-во БГУ. 2004. 63 с.
4. *Tan Y., Huang W., Sun Y., Yue Y.* Comparative Study of Different Approaches to Solve Batch Process Scheduling and Optimisation Problems // *Proceedings of the 18th International Conference on Automation & Computing*. Loughborough University. Leicestershire. UK. 2012. pp. 424–444.
5. *Adonyi R., Romero J., Puigjaner L., Friedler F.* Incorporating heat integration in batch process scheduling // *Applied Thermal Engineering*. 2003. vol. 23. pp. 1743–1762.
6. *Kreipl S.* Planning and Scheduling in supply chains: An Overview of Issues in Practice // *Production and Operations Management*. 2004. vol. 17. no. 1. pp. 77–92.
7. *Steiner G., Zhang R.* Minimizing the weight number of late jobs with Batch setup times and delivery costs on a single machine // *Vienna: Itech Education and Published*. 2007. 436 p.
8. *Koehler F., Khuller S.* Optimal Batch Schedules for Parallel Machines // *Algorithms and Data Structures: 13th International Symposium*. Berlin: Springer. 2013. pp. 475–486.
9. *Yugma C., et al.* Batching and Scheduling Algorithm for the diffusion Area in Semiconductor Manufacturing // *International Journal of Production Research*. 2012. vol. 50. no. 8. pp. 2118–2132.
10. *Chandra P.* Managing Batch Processors to reduce lead time in a semiconductor packaging line // *International Journal of Production Research*. 1997. vol. 35(3). pp. 611–633.
11. *Surjandari I., Rachman A., Dhini A.* The Batch Scheduling Model for Dynamic multiitem, Multilevel Production in an assembly Job-Shop with Parallel Machines // *International Journal of Technology*. 2015. vol. 1. pp. 84–96.
12. *Lawler E.L. et al.* Sequencing and scheduling: Algorithms and complexity // *Department of Operations Research, Statistics, and System Theory [BS]*. 1989. vol. R 8909. pp. 1–70.
13. *Петросян Л.А., Зенкевич Н.А., Семина Е.А.* Теория игр // М.: Высшая школа. 1999. 300 с.
14. *Гермейер Ю.Б.* Игры с противоположными интересами // М.: Наука. 1976. 327 с.

References

1. Hughes C., Hughes T. Parallel and distributed programming using C++. Addison-Wesley Professional. 2004. 720 p. (Russ. ed.: H'juz K., H'juz T. *Parallel'noe i raspredeleennoe programirovanie na C++*. M.: Vil'jams. 2004. 672 p.).
2. Mendez C.A. et al. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers and Chemical Engineering*. 2006. vol. 30. pp. 913–946.
3. Kovalev M.M. *Modeli i metody kalendarnogo planirovanija: Kurs lekcij* [Models and methods of scheduling. A course of lectures]. Minsk: Izd-vo BGU. 2004. 63 p. (In Russ.).
4. Tan Y. Huangi W., Sun Y., Yue Y. Comparative Study of Different Approaches to Solve Batch Process Sheduling and Optimisation Problems. Proceedings of the 18th International Conference on Automation & Computing. Loughborough University. Leicestershire. UK. 2012. pp. 424–444.
5. Adonyi R., Romero J., Puigjaner L., Friedler F. Incorporating heat integration in batch process scheduling. *Applied Thermal Engineering*. 2003. vol. 23. pp. 1743–1762.
6. Kreipl S. Planning and Scheduling in supply chains: An Overview of Issues in Practice. *Production and Operations Management*. 2004. vol. 17. no. 1. pp. 77–92.
7. Steiner G., Zhang R. Minimizing the weight number of late jobs with Batch setup times and delivery costs on a single machine. Vienna: Itech Education and Published. 2007. 436 p.
8. Koehler F. Khuller S. Optimal Batch Schedules for Parrallel Machines. Algorithms and Data Structures: 13th International Symposium. Berlin: Springer. 2013. pp. 475–486.
9. Yugma C., et al. Batching and Scheduling Algorithm for the diffusion Area in Semiconductor Manufacturing. *International Journal of Production Research*. 2012. vol. 50. no. 8. pp. 2118–2132.
10. Chandra P. Managing Batch Processors to reduce lead time in a semiconductor packaging line. *International Journal of Production Research*. 1997. vol. 35(3). pp. 611–633.
11. Surjandari I., Rachman A., Dhini A. The Batch Sheduling Model for Dynamic multiitem, Multilevel Production in an assembly Job-Shop with Parrallel Machines. *International Journal of Technology*. 2015. vol. 1. pp. 84–96.
12. Lawler E.L. et al. Sequencing and scheduling: Algorithms and complexity // Department of Operations Research, Statistics, and System Theory [BS]. 1989. vol. R 8909. pp. 1–70.
13. Petrosyan L.A., Zenkevich N.A., Semina E.A. *Teorija igr* [Game theory]. M.: Vysshaja shkola. 1999. 300 p. (In Russ.).
14. Germeier Y.B.: *Igry s neprotivopolozhnyimi interesami* [Game with no conflicting interests]. M: Nauka. 1976. 327 p. (In Russ.).

Кротов Кирилл Викторович — к-т техн. наук, доцент кафедры информационных систем, ФГАОУ ВО Севастопольский Государственный университет (СевГУ). Область научных интересов: теория расписаний, методы локальной оптимизации решений, многоуровневое программирование. Число научных публикаций — 70. krotov_k1@mail.ru; ул. Университетская, 33, Севастополь, 299053; р.т.: +7(978)730-38-19.

Krotov Kirill Victorovich — Ph.D., associate professor of information systems department, Sevastopol State University. Research interests: theory of schedules, methods of local optimization solutions, multi-level programming. The number of publications — 70. krotov_k1@mail.ru; 33, Universitetskaja str., Sevastopol, 299053; office phone: +7(978)730-38-19.

РЕФЕРАТ

***Кротов К.В.* Многоуровневая модель построения расписаний обработки партий данных в конвейерных системах при формировании комплектов и наличии ограничений.**

В настоящее время актуальными являются задачи управления вычислительными процессами в конвейерных системах. Одной из задач управления вычислительными процессами является составление расписаний обработки партий данных в конвейерной системе при наличии ограничений на интервалы времени ее функционирования и условия формирования комплектов из результатов обработки. В работе выполнена декомпозиция обобщенной цели функционирования системы построения расписаний обработки партий на совокупность взаимосвязанных иерархически упорядоченных подцелей, предполагающих определение составов партий, составов групп партий и расписаний обработки партий. В соответствии с выполненной декомпозицией и сформированными подцелями функционирования подсистем применен аппарат многоуровневого программирования для решения обобщенной задачи построения расписаний обработки партий данных. В работе реализуется обоснование многоуровневой модели (модели иерархической игры) построения комплексных расписаний обработки партий данных в конвейерной системе: обосновываются критерии оптимизации решений на каждом из уровней иерархии и ограничения на время обработки партий в системе.

SUMMARY

***Krotov K.V.* A Multilevel Scheduling Model for Data Batches Processing in Conveyor Systems when Forming Sets and in the Presence of Restriction.**

Nowadays computational process control in conveyor systems is a topical issue. Scheduling of data processing in a conveyor system with time restrictions on its operation and under condition of the formation of sets from processing results is one of the computing process management tasks. We have performed decomposition of the generalized goal of functioning of a scheduling system for data processing into a set of interrelated hierarchically arranged sub-goals that involve determining composition of data batches, groups of data batches and processing scheduling. A multilevel programming device for solving the generalized task of processing scheduling was applied in accordance with the performed decomposition and sub-goals of subsystems operation. The paper substantiates the use of the multilevel model (hierarchical game model) for complex scheduling of data processing in the conveyor system, i.e. the criteria of optimizing solutions at each hierarchical level and time restrictions on data batches processing in the system.