

П.А. Борисовский¹,

к.ф.-м.н., доцент, borisovski@mail.ru

А.В. Еремеев²

д.ф.-м.н., в.н.с., eremeev@ofim.oscsbras.ru

¹Омский государственный университет им. Ф.М. Достоевского, г. Омск

²Омский филиал Института математики им. С.Л.Соболева СО РАН, г. Омск

Построение расписаний многопродуктового производства с использованием целочисленного линейного программирования и эволюционных вычислений

Аннотация. В работе предложен новый алгоритм построения расписаний многопродуктового производства для задач большой размерности. Алгоритм основан на декомпозиционной схеме, дополненной эвристическим методом скользящего окна и генетическим алгоритмом. Вычислительный эксперимент показал преимущество предложенного алгоритма в сравнении с известным аналогом по качеству получаемых решений и времени вычислений и его применимость к реальным задачам в химической промышленности.

Ключевые слова: производственное расписание, декомпозиция, частично целочисленное линейное программирование, жадный алгоритм, генетический алгоритм, точка событий, реакция, заказ

1. Введение

При составлении расписания современного многопродуктового производства необходимо рассматривать большое число единиц оборудования, операций, промежуточных и конечных продуктов с учетом сложных взаимосвязей между ними. Задача состоит в назначении операций на единицы оборудования и выборе времени выполнения и объемов обрабатываемой

продукции в каждой из этих операций с целью по возможности полно выполнить имеющиеся заказы на продукцию в период планирования.

Известно большое число работ по моделированию такого рода задач в терминах частично целочисленного линейного программирования (ЧЦЛП) – см., например, [1-5]. Однако решение этих задач средствами ЧЦЛП связано со значительными трудностями при большом числе переменных и ограничений. С целью преодоления этих трудностей в литературе предложено несколько вариантов декомпозиционных схем [3-5], в которых период планирования разбивается на последовательность более коротких периодов (горизонтов планирования) и на каждом из этих периодов решается подзадача построения расписания, называемая далее *задачей нижнего уровня*.

В настоящей работе также используется декомпозиция и модель ЧЦЛП для задачи нижнего уровня, аналогичная модели из [5]. Для решения подзадач нижнего уровня предлагается новый эвристический метод *скользящего окна*, в котором используются идеи, аналогичные методу релаксации и фиксации [6]. С целью улучшения качества получаемых решений и сокращения времени вычислений метод скользящего окна может комбинироваться с генетическим алгоритмом, решающим вспомогательную задачу построения расписания для критических (наиболее загруженных) единиц оборудования.

2. Постановка задачи

Входные данные задачи построения расписания многопродуктового производства [3,5] состоят из

- 1) множества S видов сырья, промежуточных и конечных продуктов;
- 2) множества операций I и временных окон, когда они могут выполняться;
- 3) множества U единиц оборудования (устройств) с указанием момента, когда устройство становится доступным для использования и той операции, которая выполнялась на устройстве последней до этого момента;

4) таблицы применимости, в которой для каждой операции указана единица оборудования, на которой данная операция может выполняться (предположение о том, что каждой операции соответствует одно устройство не сужает общности постановки задачи, т.к. однотипные операции на различных устройствах могут рассматриваться как разные операции);

5) минимальный и максимальный объем партии продукта на выходе каждой операции;

6) для каждой операции указано множество потребляемых промежуточных продуктов и сырья, а также производимый продукт и коэффициенты потребления входных продуктов и сырья на единицу производимого продукта.

7) производительность каждой операции, т.е. количество продукта, получаемое в результате выполнения операции в единицу времени;

8) объемы заказов и директивные сроки, т.е. даты, не позднее которых заказана отгрузка конечного продукта, причем на один продукт может быть несколько заказов с разными директивными сроками;

9) информация о рабочих и выходных днях, начиная с первого дня периода планирования.

Задача состоит в отыскании допустимого расписания, минимизирующего целевую функцию, которая представляет собой свертку из трех критериев. С наибольшими весами в свертку входят объем недопроизводства (то есть разница между объемом заявок и объемом выпуска) и суммарное отклонение от директивных сроков. С существенно меньшим весом в свертку входит суммарное время переналадок оборудования.

В работе рассматривается производство, в котором операции по переработке сырья выполняются на сравнительно небольшом числе устройств, называемых далее *реакторами*. Кроме того, имеются вспомогательные единицы оборудования, служащие для загрузки сырья (входные загрузчики), промежуточного хранения (емкости для сырья и емкости для продукции) и отгрузки конечного продукта (погрузчики и упаковщики). Часть операций

отгрузки могут выполняться только в рабочие дни. Множество операций с такими временными окнами далее обозначается через I^w .

Каждая технология включает в себя пять операций: загрузка сырья, хранение сырья, переработка сырья на реакторе (реакция), хранение продукта и его отгрузка, совмещенная с упаковкой, если требуется упаковка. Различным способам упаковки одного и того же промежуточного продукта соответствуют разные конечные продукты.

Ввиду того, что в рассматриваемом производстве «узким местом», как правило, оказываются реакторы, особое значение имеет множество промежуточных продуктов S^p , получаемых в результате реакций.

Множество конечных продуктов S^{art} разбивается на два подмножества: продукты, выпускаемые по заказу (подмножество S^{MTO}) и выпускаемые для складирования (подмножество S^{MTS}). Весовой коэффициент при объеме недопроизводства продуктов из S^{MTO} имеет большее значение, чем для продуктов из множества S^{MTS} . Директивные сроки для продуктов из множества S^{MTS} отсутствуют и время их производства не учитываются в целевой функции.

При переходе от одной реакции к другой требуется выполнение операции переналадки реактора. Длительность операции i зависит от предшествующей и последующей реакций. Предполагается, что длительности переналадок удовлетворяют неравенству треугольника (это условие неявно предполагается и в работе [5]). Заметим, что предложенные ниже алгоритмы и модели ЧЦПП применимы и в случае, когда неравенство треугольника нарушается, однако оптимальные решения в подзадачах нижнего уровня при этом могут оказаться недостижимы. Для операций переналадки также заданы временные окна.

3. Декомпозиционный метод и задача нижнего уровня

В настоящей работе используется декомпозиционный метод, аналогичный предложенному в [5]. Период планирования разбивается на горизонты планирования и исходная задача составления расписания решается поэтапно,

как последовательность задач нижнего уровня с использованием двух моделей ЧЦЛП. В модели горизонта планирования для рабочих дней допускается использование операций отгрузки из множества I^w , а в модели горизонта планирования на выходные дни такие операции не предусматриваются. В отличие от [5], где для выделения горизонтов планирования решается оптимизационная задача верхнего уровня, в настоящей работе разбиение на горизонты планирования осуществляется непосредственно на основе исходных данных о директивных сроках заказов.

Если первый день периода планирования является рабочим, то первый горизонт состоит из рабочих дней (его длительность не превышает 5 дней). Если же первый день периода планирования выходной, то первый горизонт состоит из выходных. Далее горизонты планирования, состоящие из выходных и рабочих дней, чередуются, пока не будет исчерпан весь период планирования. В модели нижнего уровня для первого горизонта планирования учитываются только те заказы, директивные сроки которых приходятся на дни этого горизонта планирования. После решения задачи нижнего уровня для текущего горизонта планирования происходит переход к следующему горизонту, причем в рассмотрение включаются как заказы с директивными сроками в дни нового горизонта планирования, так и не выполненные в полном объеме заказы из горизонтов, рассмотренных ранее. Если некоторые устройства в новом горизонте планирования оказались заняты операциями, назначенными на предыдущих этапах декомпозиции, то информация об этих операциях передается как часть исходных в новую задачу нижнего уровня.

После решения задачи нижнего уровня для последнего горизонта планирования происходит агрегация результатов всех решенных подзадач в единое решение исходной задачи и декомпозиционный алгоритм заканчивает работу.

Модель ЧЦЛП для задачи нижнего уровня основана на *принципе непрерывного времени* [4]. Ограничения и критерий оптимизации – те же, что в

исходной задаче, но вместо всего множества заявок учитывается его подмножество, назначенное на текущий горизонт планирования.

Выбор множества выполняемых операций и порядка их выполнения осуществляется с помощью бинарных переменных wv_{in} , $i \in I$, $n \in N$, где $N = \{1, 2, \dots, N^{\max}\}$ – множество *точек событий*. Каждая точка событий соответствует некоторой позиции в последовательности выполнения операций. Переменная wv_{in} равна 1 тогда и только тогда, когда в точке событий $n \in N$ операция $i \in I$ назначена на соответствующее этой операции устройство (последнее определено однозначно). В таком случае будем говорить, что операция i активна в точке событий n . На каждом устройстве в одной точке событий может выполняться не более одной операции. Общее число точек событий N^{\max} выбирается априори на основе вычислительного эксперимента или теоретической оценки (см., например, [1]) и должно быть достаточно велико.

При $wv_{in}=1$ время начала и окончания операции i в точке событий n представлено вещественными переменными TS_{in} и TF_{in} , а вещественная переменная B_{in} определяет объем обрабатываемого вещества. В случае $wv_{in}=0$, переменные TS_{in} , TF_{in} и B_{in} могут принимать произвольные значения.

Ограничения и целевая функция рассматриваемой модели ЧЦЛП для задачи нижнего уровня совпадают с ограничениями и целевой функцией задачи нижнего уровня из [5] за исключением ограничений, относящихся к операциям переналадки реакторов, которые в настоящей работе сформулированы в модифицированном виде, представленном в [7]. Модифицированная формулировка [7] имеет преимущества при решении задач с большим числом точек событий и операций. Ввиду большого объема полное формальное описание модели здесь не приводится (см. [5,7]).

4. Метод скользящего окна и жадная эвристика для задачи нижнего уровня

Описанная в §3 задача нижнего уровня разрешима за приемлемое время современными пакетами решения задач ЧЦПП, такими как CPLEX или Express-MP, если каждый горизонт планирования содержит лишь сравнительно малое число заказов. В работе [5], например, задача нижнего уровня решается для горизонтов планирования длительностью 12 часов, каждый из которых содержит в среднем около 134 продуктов, включая промежуточные. В п.4.1 этого параграфа предлагается эвристический метод скользящего окна для более продолжительных горизонтов планирования. Далее, в п. 4.2, на основе этого метода разработан жадный алгоритм.

4.1 Метод скользящего окна

В методе скользящего окна задача нижнего уровня из §3 формулируется только для одного конечного продукта s' и в рассмотрение принимаются только те промежуточные продукты, которые могут использоваться для производства продукта s' . Число точек событий выбирается с учетом технологии выпуска выбранного конечного продукта.

Будем говорить, что переменная фиксируется с некоторым значением, если в модели ЧЦПП эта переменная заменяется постоянной величиной с указанным значением. Под снятием фиксации с переменной будем подразумевать переход к модели, в которой данная переменная снова может принимать любые значения из своей области определения.

В начальном решении полагаем $wv_{in}=0$ для всех $i \in I$, $n \in N$, т.е. ни одна операция не выполняется. Конечные продукты упорядочиваются некоторым образом и обрабатываются далее в соответствии с этой последовательностью. Для каждого конечного продукта s' выбирается некоторое множество операций $I(s') \subseteq I$ по производству промежуточных продуктов и конечного продукта s' .

Составление расписания для очередного продукта s' выполняется посредством решения серии задач ЧЦЛП из §3, в которых фиксируются бинарные переменные во всех точках событий кроме $n_s, \dots, n_e = n_s + w$, где w – настраиваемый параметр, называемый *шириной окна*. Серия начинается с задачи, в которой $n_s = 1$. При решении каждой задачи ЧЦЛП все бинарные переменные, которые не относятся к операциям из множества $I(s')$ и точкам событий n_s, \dots, n_e , фиксируются со значениями, выбранными при решении предыдущей задачи ЧЦЛП. Если в полученном решении полностью выполнен заказ на продукт s' или $n_s \geq N^{\max} - w$, то переходим к рассмотрению следующего конечного продукта. В противном случае n_s увеличивается на w и попытки выпуска продукта s' продолжаются.

4.2. Жадная эвристика

Описанный выше метод скользящего окна лежит в основе следующего жадного алгоритма.

Алгоритм 1. Жадный алгоритм

1. Положить $wv_{in} := 0$ для всех $i \in I, n \in N$.
2. Положить $P := S^{\text{art}}$, т.е. P – множество всех конечных продуктов.
3. Пока P не пусто выполнять:
 - 3.1. Выбрать конечный продукт $s \in P$ согласно жадному правилу (см. описание ниже).
 - 3.2. Положить $n_s := 1, n_e := w$.
 - 3.3. Пока объем производства продукта s меньше объема заказа на этот продукт и $n_e \leq |N|$, выполнять шаги 3.3.1-3.3.4:
 - 3.3.1. Снять фиксацию переменных wv_{in} в точках событий $n = n_s, \dots, n_e$ для операций из множества $I(s)$.
 - 3.3.2. Решить задачу ЧЦЛП нижнего уровня с учетом имеющихся фиксаций.
 - 3.3.3. Зафиксировать все бинарные переменные равными их значениям в решении задачи нижнего уровня.

3.3.4. Положить $n_s := n_s + w$; $n_e := n_e + w$.

3.4. Удалить s из P .

Жадное правило на шаге 3.1 формулируется с учетом директивных сроков и объемов заказов, причем продукты из множества S^{MTO} имеют больший приоритет, чем продукты из S^{MTS} . С этой целью сначала формируется множество продуктов-кандидатов $P' \subseteq P$ по следующему алгоритму.

Алгоритм 2. Формирование множества продуктов-кандидатов

1. Если P содержит продукты из множества S^{MTO} с директивными сроками в текущем горизонте планирования, то добавить все такие продукты в P' .
2. Если $P' = \emptyset$, то положить $P' := S^{\text{MTO}}$.
3. Если $P' = \emptyset$, а P содержит продукты из множества S^{MTS} с директивными сроками в текущем горизонте планирования, то добавить все такие продукты в P' .
4. Если $P' = \emptyset$, то положить $P' := P$.

Жадное правило состоит в выборе продукта-кандидата $s \in P'$ с наибольшим объемом заказа.

В результате фиксации большинства переменных задача ЧЦЛП в алгоритме 1 оказывается легко разрешимой современными пакетами решения задач ЧЦЛП.

Жадный алгоритм аналогичен методу релаксации и фиксации [6]. В отличие от алгоритма 1, в методе релаксации и фиксации [6] дополнительно осуществляется релаксация условий целочисленности для некоторых бинарных переменных. Однако, как показал эксперимент, релаксация условия целочисленности переменных в алгоритме 1 не приводит к существенному сокращению трудоемкости решения задачи ЧЦЛП. В связи с этим релаксация бинарных переменных в методе скользящего окна не используется.

Предложенный здесь метод скользящего окна может быть применен к задачам теории расписаний других типов, например, для производства партиями [3].

5. Построение расписания для реакторов в одном горизонте планирования

5.1. Задача построения расписания для реакторов

Сформулируем упрощенный вариант исходной задачи, рассматривая только реакторные устройства. Данная вспомогательная задача состоит в построении предварительного расписания, которое может быть использовано в гибридном алгоритме для вычисления полного решения в заданном горизонте планирования. Для дальнейшего упрощения задачи предположим, что все требуемое сырье доступно в любой момент времени в полном объеме. Введем следующие обозначения:

I^p – множество реакций ($I^p \subseteq I$);

U^p – множество реакторов ($U^p \subseteq U$);

$D_s > 0$ – объем заказа на продукт s , $s \in S^p$, отнесенный на рассматриваемый горизонт планирования;

$s_i \in S^p$ – продукт на выходе реакции $i \in I^p$;

$u_i \in U^p$ – устройство, на котором выполняется реакция $i \in I^p$;

$r_i > 0$ – производительность реакции $i \in I^p$, т.е. количество продукта s_i , производимого в единицу времени;

a_{ij} – длительность переналадки реактора u_i с реакции i на реакцию j , если $u_i = u_j$;

a'_i – длительность начальной наладки для выполнения реакции i , если эта реакция выполняется первой на реакторе u_i ;

$0 < T_i^{\max}$ – максимальная допустимая длительность реакции. В данной работе предполагается, что эта величина равна времени, которое требуется для выпуска продукции в полном объеме, т.е. $T_i^{\max} = D_{s_i} / r_i$.

Задача состоит в отыскании множества реакций, количества выпуска продукции и момента начала для каждой из реакций при указанных выше ограничениях на допустимую длительность реакций и выполнении всех заказов в полном объеме.

Целевая функция представляет собой свертку трех критериев. С наибольшим весом входит критерий общего времени выполнения всех заказов C_{\max} . С меньшими весами в целевую функцию входят критерии минимизации суммарного времени переналадок и суммарного времени использования реакторов. Веса при этих критериях обозначим через P_1 и P_2 .

5.2. Модель частично целочисленного линейного программирования для реакций в одном горизонте планирования

Обозначим через I_u множество реакций, которые могут выполняться на реакторе $u \in U^p$. Запишем модель с использованием точек событий. Пусть $K_u = \{1, 2, \dots, N_u^{\max}\}$ – множество точек событий, в которые может быть использован реактор u . С учетом неравенства треугольника, всегда существует оптимальное решение, где каждая реакция выполняется не более чем один раз. Это свойство оптимального решения будет использоваться при записи модели ЧЦЛП. В частности, из указанного свойства вытекает, что для гарантированного получения оптимального решения величину N_u^{\max} достаточно выбрать равной $|I_u|$.

Введем переменные задачи:

$x_{in} \in \{0, 1\}$ равна 1, если реакция i назначена на реактор u_i в точке событий $n \in K_u$, иначе $x_{in} = 0$;

$\delta_i > 0$ – длительность реакции i ;

α_{un} – длительность переналадки на реакторе u между точками событий n и $n+1$;

α'_u – длительность начальной наладки для выполнения реакции i , если эта реакция выполняется первой на реакторе u ;

C_{\max} – общее время завершения реакций.

Модель ЧЦЛП имеет следующий вид.

$$\min C_{\max} + P_1 \left(\sum_{i \in I^p} \delta_i \right) + P_2 \left(\sum_{u \in U^p} \left(\alpha'_u + \sum_{n \in K_u} \alpha_{un} \right) \right); \quad (1)$$

$$\sum_{i \in I_u} x_{in} \leq 1, \quad u \in U^p, \quad n \in K_u; \quad (2)$$

$$\sum_{n \in K_u} x_{in} \leq 1, \quad u \in U^p, \quad i \in I_u; \quad (3)$$

$$\sum_{i \in I_u} x_{i,n-1} \geq \sum_{i \in I_u} x_{in}, \quad u \in U^p, \quad n \in K_u, \quad n > 1; \quad (4)$$

$$\sum_{i: s_i=s} r_i \delta_i \geq D_s, \quad s \in S^p; \quad (5)$$

$$\alpha_{un} \geq \sum_{i \in I_u} a_{ij} x_{i,n-1} - M(1 - x_{jn}), \quad u \in U^p, \quad j \in I_u, \quad n \in K_u, \quad n > 1; \quad (6)$$

$$\alpha'_u \geq \sum_{i \in I_u} a'_i x_{i,1}, \quad u \in U^p; \quad (7)$$

$$\sum_{i \in I_u} \delta_i + \alpha'_u + \sum_{n \in K_u} \alpha_{un} \leq C_{\max}, \quad u \in U^p; \quad (8)$$

$$\delta_i \leq T_i^{\max} \sum_{n \in K} x_{in}, \quad i \in I^p; \quad (9)$$

$$x_{in} \in \{0,1\}, \quad i \in I^p, \quad n \in K; \quad (10)$$

$$\delta_i, \alpha_{un}, \alpha'_u \geq 0, \quad u \in U^p, \quad i \in I_u, \quad s \in S^p, \quad n \in K. \quad (11)$$

Целевая функция (1) отражает основной и второстепенный критерии. Согласно ограничениям (2) и (3), в каждой точке событий имеется не более одной реакции и любая реакция присутствует не более, чем в одной точке событий; условие (4) обеспечивает «непрерывность» использования точек событий, т.е. если на реакторе некоторая точка событий содержит реакцию, то предыдущая точка событий также занята (это свойство важно для моделирования переналадок). Неравенства (5) связывают количество произведенной продукции

и объемы заказов. Время переналадки между точками событий $n - 1$ и n задается неравенством (6). Правая часть в (6) равна a_{ij} в том и только том случае, когда $x_{i,n-1} = 1$ и $x_{jn} = 1$, где M – достаточно большая константа, например, $M = \max_{i,j} a_{ij}$. Начальные длительности наладки реакторов задаются неравенством (7). Суммарное время реакций и переналадок для каждого реактора связано с переменной C_{\max} соотношением (8). Согласно неравенству (9), длительность реакции находится в интервале $[0, T_i^{\max}]$, если реакция i выполняется; иначе $\delta_i = 0$.

Сформулированная задача является NP-трудной в сильном смысле, т.к. к ней сводится NP-трудная в сильном смысле задача о кратчайшем полуметрическом гамильтоновом пути [8].

5.3. Генетический алгоритм.

Рассмотрим адаптацию генетического алгоритма (ГА) для приближенного решения задачи (1) - (11). ГА относится к числу рандомизированных эвристик и основан на эволюционных принципах (см., например, [9]). Пробные решения задачи оптимизации представляются как *особи популяции*. Как правило, особи кодируются с помощью бинарных строк, называемых *генотипами*, однако в некоторых случаях целесообразно использовать кодировку с использованием большего числа символов. Элементы строки генотипа принято называть *генами*. Алгоритм начинает работу с некоторой популяции $\Pi^{(0)}$, представляющей собой набор генотипов, построенных случайным образом с заданным априори распределением вероятностей. На каждой итерации строится некоторое число новых пробных решений посредством применения оператора *рекомбинации* Cross и оператора *мутации* Mut к родительским решениям, выбранным из текущей популяции $\Pi^{(t)}$. Выбор родительских решений осуществляется оператором *селекции*, который действует случайным образом с вероятностным распределением, зависящим от значений *функции приспособленности* генотипов популяции $\Pi^{(t)}$. Функция приспособленности, как правило, вычисляется по значению целевой функции в том пробном

решении, которое кодирует генотип. Если генотип кодирует недопустимое решение, то функция приспособленности может вычисляться с учетом штрафа за нарушение ограничений задачи. В рассматриваемом ГА используется *стационарная стратегия управления популяцией* [10].

ГА со стационарной стратегией управления популяцией

1. Инициализировать популяцию $\mathbf{\Pi}^{(0)}$.
2. Для всех t от 1 до t_{\max} выполнять шаги 2.1-2.6:
 - 2.1 $\mathbf{\Pi}^{(t)} := \mathbf{\Pi}^{(t-1)}$.
 - 2.2 Выбрать родительские генотипы $\mathbf{p}^1, \mathbf{p}^2$ из $\mathbf{\Pi}^{(t)}$ оператором селекции.
 - 2.3 Вычислить два потомка $\mathbf{c}^1, \mathbf{c}^2$ оператором рекомбинации $\text{Cross}(\mathbf{p}^1, \mathbf{p}^2)$.
 - 2.4 Применить мутацию $\mathbf{c}^1 := \text{Mut}(\mathbf{c}^1), \mathbf{c}^2 := \text{Mut}(\mathbf{c}^2)$.
 - 2.5 Выбрать два неперспективных в некотором смысле генотипа $\mathbf{q}^1, \mathbf{q}^2$ из $\mathbf{\Pi}^{(t)}$.
 - 2.6 Заменить $\mathbf{q}^1, \mathbf{q}^2$ на $\mathbf{c}^1, \mathbf{c}^2$ в популяции $\mathbf{\Pi}^{(t)}$.

Данная стратегия управления популяцией хорошо зарекомендовала себя на практике, однако несколько отличается от используемой в классическом ГА, где на каждой итерации происходит полная замена популяции [9].

Кодировка решений. В рассматриваемом ГА генотип состоит из двух частей: перестановки $\pi = (i_1, \dots, i_{|I^p|})$ и целочисленного вектора $\mathbf{L} = (L_1, \dots, L_{|I^p|})$.

При декодировании расписания по заданному генотипу (π, \mathbf{L}) перестановка π определяет порядок просмотра реакций, а вектор \mathbf{L} – максимальное число реакций, которые могут быть назначены на каждый из реакторов.

Построение расписания начинается с решения, в котором ни одному реактору не назначено реакций. Далее последовательно просматриваются все элементы перестановки π . Пусть i – очередная реакция, выбранная в соответствии с порядком просмотра π . Если число реакций, уже назначенных на реактор $u = u_i$, не превышает L_u , то реакция i добавляется в конец последовательности операций на реакторе u_i . Иначе реакция i игнорируется.

Рассмотрим иллюстративный пример, где для реакций 1,2,3 и 4 требуется устройство 1, а для реакций 5,6,7,8 и 9 – устройство 2. Тогда генотипу $\pi = (4,7,5,2,3,9,1,8,6)$, $\mathbf{L} = (2, 3)$ соответствует назначение реакций 4,2 на устройство 1 и реакций 7,5 и 9 на устройство 2.

Процедура корректировки решений. Декодированное из произвольного генотипа решение может оказаться недопустимым, если в нем не найдется ни одной реакции по выпуску некоторых из требуемых продуктов. С целью получения допустимого решения в таком случае применяется процедура корректировки решений. В этой процедуре перестановка π просматривается вторично и в случае, если реакция i производит недостающий продукт, то эта реакция добавляется в конец последовательности операций устройства u_i , после чего соответствующий продукт помечается как произведенный и далее в процедуре корректировки решений не рассматривается.

Функция приспособленности. Значение приспособленности генотипа вычисляется как обратно пропорциональная величина к значению целевой функции (1) для пробного решения, полученного из данного генотипа с помощью процедур декодирования и корректировки. Значения переменных α_{un} и α'_u при всех $i \in I^p$, $u \in U^p$, $n \in K$ вычисляются непосредственно по последовательностям реакций, назначенных на устройства. Критерии времени завершения всех реакций и общего времени выполнения реакций оцениваются посредством решения следующей задачи линейного программирования относительно переменных δ_i , $i \in I^p$:

$$\min C_{\max}; \quad (12)$$

$$\sum_{i: s_i=s} r_i \delta_i \geq D_s, \quad s \in S^p; \quad (13)$$

$$\sum_{i \in I_u} \delta_i + \alpha'_u + \sum_{n \in K_u} \alpha_{un} \leq C_{\max}, \quad u \in U^p; \quad (14)$$

$$\delta_i \geq 0, \quad i \in I^p. \quad (15)$$

Оператор селекции и выбор удаляемых особей. В предложенном ГА используется оператор *турнирной селекции* с параметром k , так называемым *размером турнира*. При действии данного оператора из текущей популяции с равномерным распределением независимо друг от друга выбираются k генотипов и наиболее приспособленный среди них выбирается в качестве родительского. Второй родительский генотип определяется таким же способом независимо от первого.

При добавлении в популяцию генотипов потомков \mathbf{c}^1 и \mathbf{c}^2 выполняется проверка на наличие особей с такими же генотипами в текущей популяции. Если какой-то из генотипов потомков уже представлен в популяции или он имеет меньшую приспособленность чем все особи текущей популяции, то данный генотип в популяцию не добавляется.

Оператор рекомбинации. Пусть имеется два родительских генотипа $\mathbf{p}^1=(\pi^1, \mathbf{L}^1)$ и $\mathbf{p}^2=(\pi^2, \mathbf{L}^2)$ и по ним вычисляются генотипы потомков $\mathbf{c}^1=(\sigma^1, \mathbf{K}^1)$ и $\mathbf{c}^2=(\sigma^2, \mathbf{K}^2)$. Пара перестановок σ^1 и σ^2 вычисляется из π^1 и π^2 с помощью процедуры *PMX-кроссинговера* (Partially Mapped Crossover) [11], а строки \mathbf{K}^1 и \mathbf{K}^2 получаются из \mathbf{L}^1 и \mathbf{L}^2 с помощью процедуры равномерного кроссинговера (Uniform Crossover) [10].

В процедуре PMX-кроссинговера два индекса l и r , $l < r$, выбираются с равномерным распределением и компоненты перестановок π^1 и π^2 от l до r копируются в перестановки потомков σ^2 и σ^1 с обменом: $\sigma_j^2 = \pi_j^1$, $\sigma_j^1 = \pi_j^2$, $j=l, \dots, r$. Далее, для каждого $j \notin \{l, \dots, r\}$ проверяется, можно ли в этой позиции оставить значение из родительского решения, а именно, если π_j^1 отсутствует среди скопированных генов, то полагаем $\sigma_j^1 = \pi_j^1$. Аналогично модифицируется σ^2 . Остальные компоненты $j \notin \{l, \dots, r\}$ заполняются с помощью отображений $M^1(\pi_j^2) = \pi_j^1$ и $M^2(\pi_j^1) = \pi_j^2$. В каждом гене $j \notin \{l, \dots, r\}$ строки σ^1 полагаем $\sigma_j^1 = M^1(\pi_j^1)$, если значение $M^1(\pi_j^1)$ еще отсутствует в σ^1 ; иначе $\sigma_j^1 = M^1(M^1(\pi_j^1))$, если значение $M^1(M^1(\pi_j^1))$ еще отсутствует в σ^1 , и так далее.

Вторая перестановка заполняется аналогично с использованием отображения M^2 . Иллюстративный пример выполнения РМХ-кроссинговера приведен на рис. 1.

$$\begin{array}{r}
 \pi^1 = (1 \quad 2 \quad 3 \mid 4 \quad 5 \quad 6 \quad 7 \mid 8 \quad 9) \\
 \pi^2 = (4 \quad 5 \quad 2 \mid 1 \quad 8 \quad 7 \quad 6 \mid 9 \quad 3) \\
 \\
 \\
 \sigma^1 = (4 \quad 2 \quad 3 \mid 1 \quad 8 \quad 7 \quad 6 \mid 5 \quad 9) \\
 \sigma^2 = (1 \quad 8 \quad 2 \mid 4 \quad 5 \quad 6 \quad 7 \mid 9 \quad 3).
 \end{array}$$

Рис. 1: Оператор РМХ-кроссинговера.

В процедуре равномерного кроссинговера для каждого $u \in U^p$ с вероятностью 0.5 независимо от других генов полагаем $K_u^1 := L_u^1$, $K_u^2 := L_u^2$; в противном случае полагаем $K_u^1 := L_u^2$, $K_u^2 := L_u^1$.

Оператор мутации. В операторе мутации к перестановке π последовательно применяются следующие три процедуры. В процедуре *обмена* из множества $\{1, \dots, |I^p|\}$ равновероятно выбираются два номера генов j и j' , после чего выполняется обмен значений в генах π_j и $\pi_{j'}$. В процедуре *вставки* равновероятно выбирается один номер гена $j \in \{1, \dots, |I^p|\}$ и ген π_j переносится в новую случайно выбранную позицию j' . Прежние значения в генах между j и j' (включительно) сдвигаются на одну позицию. Процедура *удаления* состоит в выборе наиболее загруженного реактора и удалении с него одной равновероятно выбранной реакции.

Мутация вектора \mathbf{L} выполняется следующим образом: каждый ген L_u , $u=1, \dots, m$, такой что $0 < L_u < |I_u|$, увеличивается на 1 с вероятностью p_m , либо уменьшается на 1 с вероятностью p_m . В противном случае, т.е. с вероятностью $1-2p_m$, ген L_u остается без изменений. Если же $L_u=0$, то этот ген может только увеличиваться, что происходит с вероятностью p_m . Если же $L_u=|I_u|$, то этот ген может только уменьшаться с вероятностью p_m .

Полученный в результате мутации генотип может давать недопустимое решение при декодировании – в таком случае к генотипу применяется процедура корректировки решений и модифицированный генотип добавляется в популяцию.

6. Гибридный алгоритм для задачи нижнего уровня

Предлагаемый гибридный алгоритм основан на эвристике скользящего окна, применяемой с учетом решения, найденного генетическим алгоритмом. В процессе работы декомпозиционного алгоритма в каждом горизонте планирования строится своя подзадача ЧЦЛП для реакций в соответствии с моделью из п. 5.2. Данная задача решается приближенно с помощью генетического алгоритма, далее применяется эвристика скользящего окна, вычисляющая более детальное решение в соответствии с моделью из §3. При работе эвристики скользящего окна последовательность выполнения заказов и назначение операций на реакторы соответствуют расписанию реакций, построенному генетическим алгоритмом. Гибридный алгоритм имеет следующий вид.

Гибридный алгоритм

1. Построить с помощью ГА расписание для реакций на текущий горизонт планирования.
2. Упорядочить реакции по моменту начала их выполнения в решении, полученном ГА. Пусть $i_1, \dots, i_{k'}$ – полученная последовательность реакций, $s_1, \dots, s_{k'}$ – последовательность конечных продуктов, соответствующих этим реакциям и $d_1, \dots, d_{k'}$ – объемы производства конечных продуктов.
3. Сформулировать задачу нижнего уровня с множеством конечных продуктов $\{s_1, \dots, s_{k'}\}$ и объемами заказов $d_1, \dots, d_{k'}$. Директивные сроки продуктов соответствуют данным исходной задачи.

4. Методом скользящего окна построить решение задачи в текущем горизонте планирования, просматривая конечные продукты в последовательности s_1, \dots, s_k и полагая $I(s_j) = \{i_j\}, j=1, \dots, k$.

На рис. 2 и 3 приведены диаграммы Гантта для решений, полученных ГА и гибридным алгоритмом в случае 10 реакторов M1-M10.

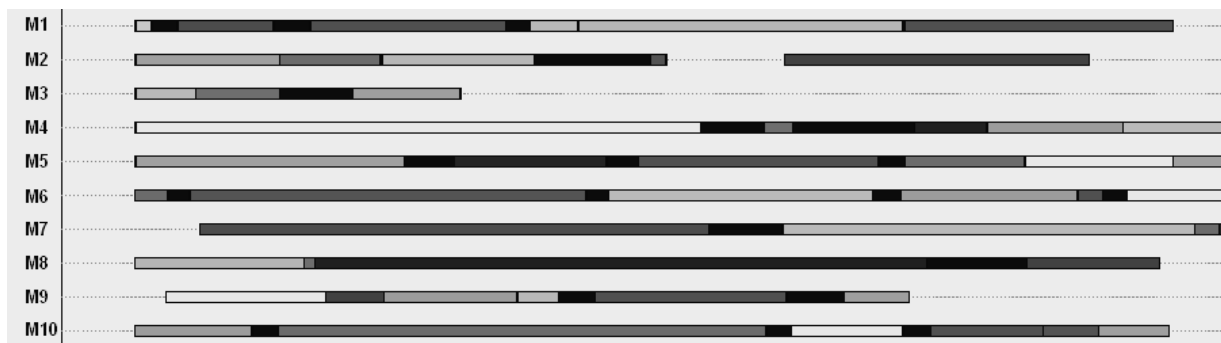


Рис 2: Решение подзадачи для реакций, найденное генетическим алгоритмом

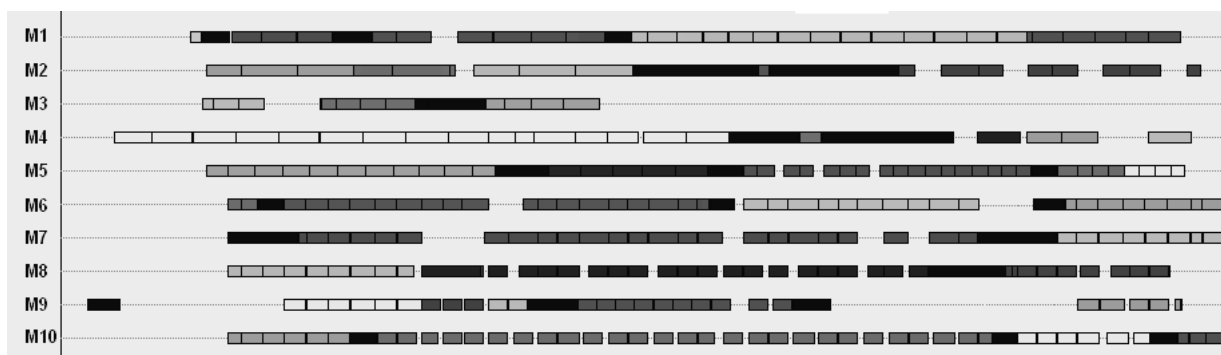


Рис 3: Решение задачи нижнего уровня, найденное гибридным алгоритмом

Решение, вычисленное гибридным алгоритмом, включает в себя также расписание для загрузчиков, емкостей хранения и отгрузки, однако на рис. 3 изображены только реакторы. Как видно из рисунков, решение гибридного алгоритма отличается меньшей длительностью реакций и большим их числом. Это связано с ограниченностью емкостей для складирования продуктов реакций. Всякий раз при наполнении этих емкостей необходимо окончить реакцию и отгрузить продукт. Другое отличие состоит в том, что на рис. 2 имеется только один свободный промежуток между реакциями, в то время как на рис. 3 между реакциями имеются многочисленные промежутки. Это различие связано с тем, что в модели из §3 учитываются директивные сроки, в

то время как в модели из п. 5.2 основным критерием является минимизация общего времени завершения операций. Кроме того, в некоторые промежутки времени операции по отгрузке могут быть запрещены, что также ведет к отсрочке некоторых реакций и/или сокращению их объема.

7. Вычислительный эксперимент

В настоящем параграфе проводится сравнение предложенных алгоритмов с коммерческим пакетом решения задач ЧЦЛП CPLEX 12.3 и с декомпозиционным алгоритмом из [5]. В п. 7.1 генетический алгоритм из п. 5.3 сравнивается с пакетом CPLEX при решении задачи (1)-(11). Далее в п. 7.2 тестируется декомпозиционный метод (см. §3) при использовании гибридного либо жадного алгоритмов решения задачи нижнего уровня. Эксперименты выполнялись на ЭВМ с процессором AMD Phenom 2.8 МГц при использовании одного ядра процессора. Исключение составляет процедура решения задач ЧЦЛП в пакете CPLEX, где допускалось использование двух ядер.

7.1. Сравнение генетического алгоритма с пакетом CPLEX

Генетический алгоритм был запрограммирован на языке C++ в системе MS Visual Studio 6.0. При решении задач линейного программирования (12)–(15), использовалась реализация двойственного симплекс-метода из пакета программ QSOPT. Кроме того, модель (1)-(11) была представлена в системе GAMS, интегрированной решателем задач ЧЦЛП из пакета CPLEX 12.3.

Тестовые примеры были построены случайным образом со следующими параметрами: $D_s \in [10; 20]$ для всех $s \in S^p$, $r_i \in [1; 20]$ для всех $i \in P$, $a_{ij} \in [0; 30]$ для всех $i \in P$ и $j \in P$, таких что $u_{i_s} = u_j$. Для каждой реакции $i \in P$ выбор устройства u_i , на котором она может выполняться, осуществлялся равновероятно на множестве U^p . Соответствие реакций продуктам устанавливалось таким образом, чтобы каждый продукт выпускался хотя бы одной из реакций: операции с порядковым номером не более $|S^p|$ назначались на

выпуск продукта с тем же порядковым номером; для операций с бóльшим номером продукт выбирался равновероятно из S^p . Мощность множеств S^p, U^p и I^p , а также число бинарных переменных в модели ЧЦЛП и верхняя граница на время счета алгоритмов приводятся в табл. 1.

задача	$ S^p \times U^p \times I^p $	бинарных переменных	ограничение по времени (с)
1	30×2×80	1885	300
2	40×2×150	5476	600
3	50×3×180	6604	900
4	60×3×200	8642	1200
5	150×10×500	21950	3600

Таблица 1. Параметры задач и ограничение на время счета

Настраиваемые параметры ГА выбраны следующим образом: численность популяции равна 400, размер турнира $k=20$, вероятность мутации $p_m=0.05$. Второстепенные критерии входят в целевую функцию задачи (1)-(11) с весами $P_1 = P_2 = 0.001$.

На каждом тестовом примере пакет CPLEX и ГА выполнялись в течении равных промежутков времени, выбранных как указано в табл. 1. Генетический алгоритм за отведенное время выполнялся 5 раз с независимой случайной инициализацией начальной популяции (т.е. время счета при каждом запуске ГА составляло 1/5 от указанного времени). Лучшее из решений, полученных за пять запусков, считалось окончательным ответом ГА. Результаты приведены в табл. 2: общее время завершения «Стах», суммарное время переналадок «вр. переналад.» и суммарное время выполнения операций «вр. реакций».

задача	CPLEX			ГА		
	Стах	время переналад.	время реакций	Стах	время переналад.	время реакций
1	143	78	209	141	64	219
2	209	153	264	183	73	293

3	173	208	311	162	123	357
4	275	356	469	229	155	524
5	202	939	1051	183	492	1291

Таблица 2: Сравнение решений, полученных ГА и CPLEX

Как видно из табл. 2, в целом результаты ГА превосходят результаты пакета CPLEX. Наиболее существенное преимущество наблюдается по общему времени переналадок. Несмотря на то, что общее время реакций меньше в решениях, полученных пакетом CPLEX, легко видеть, что по сумме времени выполнения реакций и переналадок решения ГА имеют преимущество.

7.2. Исследование декомпозиционного алгоритма на задаче большой размерности

Далее рассматривается задача с реальными данными из [5], в которой имеется 10 реакторов, порядка 100 вспомогательных устройств и более 100 заказов на период планирования, равный одному месяцу. Как указано в §6, период планирования разбивался на горизонты из рабочих дней и горизонты из выходных дней. Максимальное число точек событий $N^{\max}=50$ для горизонтов из рабочих дней, а для горизонтов из выходных $N^{\max}=20$.

	Алгоритм декомпозиции [5]	Декомпозиция с жадным алгоритмом	Декомпозиция с гибридным алгоритмом
недопроизводство	0.96%	0.02%	0.07%
число переналадок	324	194	115
время переналадок	930	851	490
время счета	16ч	12ч	6ч

Таблица 3: Сравнение результатов для задачи большой размерности

Предложенные алгоритмы тестировались в двух вариантах: в первом случае задача нижнего уровня решалась жадной эвристикой, описанной в §4; во втором случае для этого использовался гибридный алгоритм из §6. В обоих

вариантах в методе скользящего окна при каждом обращении к CPLEX решателю отводилось 90 секунд. Время счета ГА ограничивалось 5 минутами для горизонтов из выходных дней и 10 минутами для рабочих дней. Полученные результаты приведены в табл. 3. Здесь указаны следующие параметры решений: недопроизводство, общее число переналадок, общая длительность переналадок и время работы алгоритма. Лучшие значения выделены жирным шрифтом.

Для сравнения в табл. 3 также приводятся результаты, полученные алгоритмом декомпозиции из [5] при использовании горизонтов планирования длительностью 12 часов. Эти результаты несколько отличаются от опубликованных в [5] из-за использования более новой версии CPLEX и более производительной ЭВМ в настоящей работе.

Все три алгоритма нашли решения, в которых директивные сроки для продуктов из множества S^{MTO} выполняются. Как видно из табл. 3, во всех полученных решениях недопроизводство незначительно. По числу переналадок и их длительности предложенные алгоритмы имеют преимущество по сравнению с алгоритмом из [5]. Время счета также сократилось.

Кроме того были проведены дополнительные эксперименты, где в задаче построения расписания для реакторов вместо общего времени завершения C_{max} использовался критерий минимизации недопроизводства на текущем горизонте планирования при дополнительном ограничении, обеспечивающем окончание всех операций в течении горизонта планирования. Было установлено, что данная модификация уступает модели, предложенной в п. 5.1 по количеству переналадок и равномерности загрузки оборудования.

8. Заключение

В настоящей работе получил дальнейшее развитие предложенный К. Флудасом с соавторами подход к решению задач построения производственных расписаний с использованием декомпозиции по времени и

частично-целочисленного линейного программирования. С целью повышения качества решений и сокращения времени счета предложены гибридный и жадный алгоритмы решения задачи нижнего уровня. Гибридный алгоритм основан на комбинации разработанного авторами метода скользящего окна с генетическим алгоритмом.

На задаче с реальными данными большой размерности использование метода скользящего окна позволило увеличить длительность горизонта планирования, рассматриваемого на одном этапе декомпозиции, с 12 часов до нескольких суток. Сравнение результатов полученного алгоритма декомпозиции с результатами известного ранее декомпозиционного алгоритма показало преимущество нового алгоритма.

Метод скользящего окна имеет достаточно общий вид и может быть применен к другим задачам теории расписаний, записываемым в терминах частично целочисленного линейного программирования с использованием непрерывного времени и точек событий.

Работа поддержана РФФИ (проекты 12-01-00122 и 13-01-00862).

Список литературы

[1] Еремеев А.В., Коваленко Ю.В. О задаче составления расписаний с группировкой машин по технологиям // Дискретный анализ и исследование операций. 2011. Т. 18, N 5. С. 54-79.

[2] Cho D. A solution approach to the design of multi-period, multipurpose batch plants // Korean Journal of Chemical Engineering. 2002. Т. 19, N 2. P. 197-205.

[3] Janak S. L., Floudas C. A., Kallrath J., Vormbrock N. Production scheduling of a large-scale industrial batch plant. I. Short-term and medium-term scheduling // Industrial & Engineering Chemistry Research. 2006. Vol. 45, N 25. P. 8234-8252.

[4] Ierapetritou M.G., Floudas C.A. Effective continuous-time formulation for short-term scheduling: I. Multipurpose batch processes // Industrial & Engineering Chemistry Research. 1998. Vol. 37. P. 4341-4359.

[5] Shaik M.A., Floudas C.A., Kallrath J., Pitz H.-J. Production scheduling of a large-scale industrial continuous plant: Short-term and medium-term scheduling // Computers & Chemical Engineering. 2009. Vol. 33. P. 670-686.

[6] Pochet Y., Wolsey L.A. Production Planning by Mixed Integer Programming. Series in Operations Research and Financial Engineering. Heidelberg: Springer, 2006. 500 p.

[7] Borisovsky P., Ereemeev A., Kallrath J. Reducing the number of changeover constraints in a MIP formulation of a continuous-time scheduling problem // To appear in Proc. of VIII International School-Symposium “Analysis, Modelling, Management and Development of Economical Systems” (AMUR’2014). – Sevastopol: DEN, 2014. URL: <http://arxiv.org/abs/1408.5832> (дата обращения 26.08.2014).

[8] Itai A., Papadimitriou C.H., Szwarcfiter J.L. Hamilton paths in grid graphs // SIAM Journal on Computing. 1982. Vol. 11, N 4. P. 676-686.

[9] Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning. Reading: Addison Wesley, 1989. 412 p.

[10] Reeves C.R. Genetic algorithms for the operations researcher // INFORMS Journal on Computing. 1997. Vol. 9, N 3. P. 231-250.

[11] Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Berlin: Springer-Verlag, 1996. 387 p.