ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель–июнь 2019. Т. 26, № 2. С. 60–78

УДК 519.8 DOI 10.33048/daio.2019.26.634

АЛГОРИТМ ЛОКАЛЬНОГО ПОИСКА ДЛЯ ПОСТРОЕНИЯ РАСПИСАНИЙ РАБОТЫ ОДНОГО СТАНКА С ПЕРЕНАЛАДКОЙ ОБОРУДОВАНИЯ И СКЛАДОМ*)

П. А. Кононова[1,2,a], *Ю. А. Кочетов*[1,2,b]

[1]Институт математики им. С. Л. Соболева СО РАН,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия

[2]Новосибирский гос. университет,
ул. Пирогова, 1, 630090 Новосибирск, Россия

E-mail: [a]pkononova@math.nsc.ru, [b]jkochet@math.nsc.ru

Аннотация. Получена новая математическая модель оптимизации расписаний работы одного станка, описывающая реальное производство тротуарной плитки. Модель учитывает технологические задержки при переходе от одного типа продукции к другому, минимальные объёмы производства, разнородность заказов клиентов и наличие запасов на складе. В качестве критерия оптимизации выступает штраф за опоздание заказов относительно директивных сроков и суммарная стоимость хранения готовой продукции на складе. Построена модель частично-целочисленного линейного программирования, позволяющая решать задачи небольшой размерности. Для реальных задач месячного планирования разработан вероятностный метод локального поиска с запретами. Приводятся результаты численных экспериментов на тестовых примерах одной компании Новороссийска. Табл. 3, ил. 1, библиогр. 25.

Ключевые слова: поиск с запретами, расписание, директивный срок, запаздывание, переналадка оборудования.

Введение

Задачи построения расписаний одного станка имеют большую историю и огромное число публикаций. Первые работы в этой области появились еще в 1955–56 гг. [17, 24]. Часть задач оказалась полиномиально разрешимой, но многие практически важные задачи NP-трудны, в частности, многие задачи одного станка с переналадкой оборудования [8].

*)Работа выполнена при поддержке Программы фундаментальных научных исследований СО РАН № I.5.1 (проект № 0314–2019–0014).

Подробный обзор результатов в этой области можно найти в [7, 11, 23]. Близкие по смыслу производственные модели рассматривались в [12,18].

В настоящей работе исследуется новая прикладная задача одного станка с переналадкой оборудования, возникающая при моделировании производства тротуарной плитки методом полусухого вибропрессования. При построении расписаний приходится учитывать как особенности данного производства, в частности, минимально допустимые объёмы производства, склад готовой продукции и наличие определённых запасов на складе, так и особенности работы компании с клиентами. Получаемые компанией заказы имеют директивные сроки выполнения и разнородный состав продукции по номенклатуре. Нарушение директивных сроков поставки приводит к выплате штрафов. Они увеличиваются с ростом опозданий. Хранение готовой продукции на складе также приводит к дополнительным затратам. Построение расписаний для минимизации суммарных затрат является новой интересной задачей. Для её решения получена математическая модель частично-целочисленного линейного программирования. Она позволяет находить оптимальные решения при небольшом плановом периоде и малом числе заказов. Однако при месячном планировании такой подход оказывается неэффективным. Для этого случая разработан приближённый алгоритм локального поиска с запретами. Предполагается, что решения можно кодировать перестановками партий продукции, что соответствует реальному планированию данного производства. На таком классе примеров алгоритм легко находит оптимальное решение при малом числе партий, до 30. На больших примерах месячного планирования алгоритм за 10 минут порождает приближённые решения, значительно превосходящие по качеству решения пакета Gurobi, получаемые им за час работы.

Статья организована следующим образом. В разд. 1 приводится содержательная постановка задачи. В разд. 2 вводятся обозначения и даётся точная математическая формулировка задачи с непрерывными и целочисленными переменными. Эта формулировка имеет нелинейные ограничения для склада и порядка выполнения работ, но введением дополнительных переменных и ограничений удаётся получить задачу линейного частично-целочисленного программирования. Этот подход открывает путь к точному решению задачи, например, пакетом Gurobi. В разд. 3 излагается общая схема локального поиска с запретами для приближённого решения задач реальной размерности. В разд. 4 обсуждаются результаты численных экспериментов и проводится сравнение приближённого алгоритма с пакетом Gurobi. Разд. 5 подводит итог проделанной работе, показывает нерешённые проблемы и возможности для дальнейших исследований.

1. Содержательная постановка задачи

Имеется станок, предназначенный для производства разнотипной продукции. Для каждого типа продукции известны производительность станка, т. е. количество продукции за час работы, и её минимально допустимый объём, меньше которого нельзя производить из-за технологических особенностей производства. При переходе от одного типа продукции к другому требуется переналадка оборудования. Длительность переналадки зависит от обоих типов и может быть различной для разных пар. Для хранения готовой продукции имеется склад неограниченной вместимости. Известна удельная стоимость хранения продукции. Предполагается, что в начальный момент времени на складе имеется определённый запас продукции каждого типа.

Клиенты делают заказы на производство продукции. Заказ состоит из одной или нескольких партий продукции, например 100 штук красной плитки и 200 штук белой плитки заданных размеров (заказ из двух партий определённых типов). Клиент определяет объём требуемой ему партии по своему усмотрению. Таким образом, каждому заказу задаётся директивный срок его выполнения и объёмы партий продукции различного типа. При выполнении заказа продукция либо берётся со склада, полностью или частично, либо производится без прерываний в объёме не меньше минимально допустимого объёма и отправляется на склад. Заказ считается выполненным, когда вся заказанная клиентом продукция оказалась на складе. Если этот момент наступает позже директивного срока, то выплачивается штраф за задержку заказа. Штраф прямо пропорционален длительности задержки. Требуется найти такое расписание работы станка, чтобы минимизировать суммарный штраф за возможные задержки заказов и суммарную плату за хранение готовой продукции на складе.

2. Математическая постановка задачи

2.1. Входные данные. Обозначим через K множество типов выпускаемой продукции, J — множество заказов клиентов, I — множество партий продукции для всех заказов. Множество $I(k) \subseteq I$ задаёт все партии продукции типа k, а номер $k(i)$ указывает на тип продукции для партии i. Кроме того, номер $j(i)$ определяет заказ, к которому относится эта партия, а множество I_j задаёт все партии заказа j. Для каждого типа продукции $k \in K$ известны минимально допустимый объём производства q_k и количество производимой продукции в час p_k. Неотрицательные величины $\Delta_{kk'}$ определяют время переналадки станка при переходе с продукции типа k на k'. Без ограничения общности можно предполагать, что эти величины удовлетворяют неравенству треугольника,

т. е. $\Delta_{kk'} \leqslant \Delta_{kk''} + \Delta_{k''k'}$ для всех троек $k, k', k'' \in K$. Действительно, если это не так для некоторой тройки индексов (k, k', k''), то при переходе с типа k на k' всегда можно сначала наладить станок на тип k'', а затем на тип k' для сокращения общего времени переналадки. Данное неравенство будет использоваться ниже для построения математической модели. Кроме того, будем считать, что $\Delta_{kk} = 0$ для всех $k \in K$. Через d_j обозначим директивный срок выполнения заказа j, а через v_i — размер партии i.

Склад неограниченной вместимости в начальный момент времени содержит неотрицательный запас h_k продукции каждого типа $k \in K$. Эти запасы могут использоваться во время производства для сокращения размера партий. Пополнение запасов может происходить только излишками продукции, если такие излишки появляются в ходе производства при размере партии меньше минимально допустимого объёма производства q_k, $k \in K$. Производить продукцию впрок запрещено, т. е. если при выполнении партии i необходимо производить продукцию типа k, то её нельзя произвести в большем объёме, чем величина $\max(q_k, v_i)$. Если для партии i' тоже необходима продукция типа k, то их можно выполнять последовательно, чтобы избежать переналадки оборудования.

Через β_i обозначим стоимость хранения на складе партии i в единицу времени, а через α_j — штраф за опоздание заказа j на единицу времени.

2.2. Нелинейная модель. Введём переменные задачи:

$T_j \geqslant 0$ — задержка заказа j относительно директивного срока,

$C_j \geqslant 0$ — время окончания производства всех партий заказа j,

$s_i \geqslant 0$ — начало производства партии i,

$c_i \geqslant 0$ — окончание производства партии i,

$e_i \geqslant 0$ — время хранения партии i на складе до момента отгрузки соответствующего заказа клиенту,

$u_{ii'} \in \{0, 1\}$ — порядок выполнения партий: $u_{ii'} = 1$, если партия i выполняется до партии i', и $u_{ii'} = 0$ в противном случае,

$x_i \in \{0, 1\}$ — индикатор производства партии i: $x_i = 1$, если партия i производится хотя бы частично, и $x_i = 0$ в противном случае,

$z_i \geqslant 0$ — доля партии i, которая производится на станке: $z_i < 1$, если партию i частично брали со склада, и $z_i > 1$, если $v_i < q_{k(i)}$,

$y_i \geqslant 0$ — доля партии i, которая берётся со склада.

С использованием введённых обозначений задача минимизации суммарного штрафа за опоздание заказов и стоимости хранения готовой продукции на складе может быть записана следующим образом.

Задача 1. *Найти минимум целевой функции*

$$F = \sum_{j \in J} \alpha_j T_j + \sum_{i \in I} \beta_i e_i \tag{1}$$

при ограничениях

$$c_i = s_i + p_i z_i, \quad i \in I, \tag{2}$$

$$u_{ii'} + u_{i'i} = 1, \quad i \neq i',\ i, i' \in I, \tag{3}$$

$$s_i \geqslant c_{i'} + \Delta_{k(i')k(i)} x_i x_{i'} - M u_{ii'}, \quad i \neq i',\ i, i' \in I, \tag{4}$$

$$C_{j(i)} \geqslant c_i, \quad i \in I, \tag{5}$$

$$T_j \geqslant C_j - d_j, \quad j \in J, \tag{6}$$

$$e_i \geqslant d_{j(i)} - c_i, \quad i \in I, \tag{7}$$

$$e_i \geqslant C_{j(i)} - c_i, \quad i \in I, \tag{8}$$

$$z_i + y_i \geqslant 1, \quad i \in I, \tag{9}$$

$$z_i v_i \geqslant q_{k(i)} x_i, \quad i \in I, \tag{10}$$

$$M_i x_i \geqslant z_i, \quad i \in I, \tag{11}$$

$$\sum_{i' \in I(k)} v_{i'}(z_{i'} - 1) u_{i'i} + h_{k(i)} \geqslant v_i y_i, \quad i \in I, \tag{12}$$

$$x_i \in \{0, 1\}, \quad u_{ii'} \in \{0, 1\}, \quad i, i' \in I, \tag{13}$$

$$y_i \geqslant 0,\ z_i \geqslant 0,\ s_i \geqslant 0,\ c_i \geqslant 0,\ e_i \geqslant 0,\ T_j \geqslant 0,\ C_j \geqslant 0, \quad i \in I,\ j \in J. \tag{14}$$

Целевая функция (1) задаёт суммарный штраф за опоздание заказов и стоимость хранения готовой продукции на складе. Равенства (2) определяют время завершения производства каждой партии в зависимости от времени старта работ и их объёма. Заметим, что если партия i целиком берётся со склада, то $c_i = s_i$, а величина s_i определяется из неравенств (4). Равенства (3) вместе с неравенствами (4) гарантируют линейный порядок выполнения всех партий. Большая константа M используется здесь для того, чтобы сделать это ограничение неактивным в случае $u_{ii'} = 1$, т. е. когда партия i выполняется раньше партии i'. Так как это неравенство должно выполняться для любой пары i и i', в качестве константы M может быть взята оценка сверху на длину оптимального расписания, например, $M = \max\limits_{i \in I} c_i$ для некоторого допустимого решения задачи. Заметим, что неявно в условиях (4) используется неравенство треугольника, поскольку в противном случае это условие некорректно и может создавать искусственные задержки при переходе с одной партии на другую. Неравенства (5) и (6) определяют время завершения заказов и их возможную задержку относительно директивных сроков. Неравенства (7) и (8) позволяют получить время ожидания партий на складе до момента их отправки клиентам. Заметим, что эта величина отличается от хорошо известного в теории расписаний опережения директивных сроков (earliness), так как учитывает возможное опоздание заказов. Неравенства (9) гарантируют производство партий целиком или частично за счёт запасов на складе. Условия (10) требуют завышения объёма

производства, если он меньше минимально допустимого. Условия (11) связывают дискретные переменные x_i с непрерывными переменными z_i, позволяя определить, запускалось ли производство партий и, как следствие, какая должна быть переналадка оборудования в условиях (4). Величина M_i здесь определяется как $M_i = \max[1, q_{k(i)}/v_i]$, $i \in I$, и задаёт верхнюю оценку на рост объёма выпуска продукции до минимально допустимого. Наконец, условия (12) гарантируют наличие на складе требуемых запасов продукции с учётом их расходования и пополнения. Правая часть неравенства задаёт требуемый объём продукции типа $k(i)$ к моменту начала производства партии i. Левая часть неравенства определяет наличие такой продукции в виде исходного запаса и объёмов его расхода–пополнения к этому моменту времени. Условия (13), (14) задают область изменения переменных.

2.3. Линеаризация модели. В представленной математической модели ограничения (4) и (12) нелинейные. Для линеаризации ограничений (4) введём новые булевы переменные $x_{ii'}$, $i, i' \in I$, смысл которых определяется следующим равенством:

$$x_{ii'} = x_i x_{i'}, \quad i, i' \in I.$$

Легко проверить, что выполнение этого равенства гарантируется следующими линейными неравенствами:

$$x_{ii'} \leqslant x_i, \quad i, i' \in I,$$

$$x_{ii'} \leqslant x_{i'}, \quad i, i' \in I,$$

$$x_{ii'} \geqslant x_i + x_{i'} - 1, \quad i, i' \in I.$$

Тогда ограничение (4) можно заменить таким линейным ограничением:

$$s_i \geqslant c_{i'} + \Delta_{k(i')k(i)} x_{ii'} - M u_{ii'}, \quad i, i' \in I.$$

Аналогично для линеаризации ограничения (12) введём новые переменные

$$z_{ii'} = z_i u_{ii'}, \quad i, i' \in I,$$

и заменим (12) линейным ограничением

$$\sum_{i' \in I(k)} v_{i'}(z_{i'i} - u_{i'i}) + h_{k(i)} \geqslant v_i y_i, \quad i \in I,$$

и тремя дополнительными неравенствами, гарантирующими корректность такой замены:

$$z_{ii'} \leqslant M_i u_{ii'}, \quad i, i' \in I,$$

$$z_i - M_i(1 - u_{ii'}) \leqslant z_{ii'} \leqslant z_i, \quad i, i' \in I,$$

$$z_{ii'} \geqslant 0, \quad i, i' \in I.$$

Полученную модель частично-целочисленного линейного программирования можно использовать для решения поставленной задачи в случае небольших размерностей, используя, например, пакет Gurobi.

3. Локальный поиск на перестановках

Так как станок выполняет одну партию за другой в некотором порядке, логично кодировать решения в виде перестановок σ номеров партий от 1 до $n = |I|$. Для получения приближённого решения будем предполагать, что запас продукции на складе используется жадным образом. Перед производством очередной партии выполняется проверка, есть ли необходимая продукция на складе. Если есть, то она используется для сокращения размера партии. Если для очередной партии i запас на складе меньше размера партии v_i, то выполняется переналадка оборудования и производится недостающее количество продукции в объёме не меньше минимально допустимого, т. е. $z_i > 0$ и $x_i = 1$. Если же запас не меньше v_i, то переналадка станка не производится, партия считается выполненной и $z_i = x_i = 0$. В обоих случаях можно вычислить время старта s_i и окончания данной партии c_i, а также длительность переналадки станка в зависимости от того, какой тип продукции выпускался непосредственно перед данной партией. Такая схема использования склада взята из практики и направлена на максимально быстрое использование запасов для сокращения размеров партий и экономии на переналадке оборудования. Она хороша, когда длительности переналадки достаточно малы по сравнению со временем выполнения партий и не сильно отличаются друг от друга. В противном случае может потребоваться другая эвристика.

Вычисляя время старта для каждой партии, мы предполагали, что после выполнения одной партии сразу начинает выполняться следующая. Станок не простаивает: на нём выполняется переналадка или идёт производство очередной партии. Такие расписания принято называть активными [10]. Однако наличие второго слагаемого в целевой функции может приводить к простоям станка при высокой плате за использование склада $\beta_i \gg \alpha_{j(i)}$, $i \in I$. Ниже представлена процедура декодирования, которая по перестановке σ и жадной схеме использования склада находит точные значения переменных s_i с учётом возможных простоев станка.

3.1. Процедура декодирования. Зная значения переменных x_i, z_i, $u_{ii'}$, $i, i' \in I$, определим значения переменных s_i по условиям (2), (4):

$$s_i = \max_{i' \in I} \{ s_{i'} + p_{i'} z_{i'} + \Delta_{k(i')k(i)} x_i x_{i'} \mid u_{ii'} = 0 \}, \quad i \in I.$$

Получим активное расписание, которое не позволяет станку простаивать.

Предположим, что последняя партия в перестановке имела номер i_0, выполнялась на станке и закончилась раньше директивного срока, т. е. $c_{i_0} < d_{j(i_0)}$. Тогда простой станка на величину $\tau = d_{j(i_0)} - c_{i_0}$ перед выполнением этой партии приводит к падению целевой функции (1) на величину $\beta_{i_0}\tau$. Партия i_0 не хранится на складе, а сразу отправляется клиенту. Дальнейший простой станка приводит не только к задержке заказа, но и к росту платы за хранение всех партий этого заказа, кроме последней.

Предположим теперь, что две партии i_0 и i_1 разных заказов выполнялись последними в перестановке, $\sigma(i_1) < \sigma(i_0)$, станок не простаивал и $d_{j(i_0)} \leqslant c_{i_0} \leqslant d_{j(i_1)}$, но $\beta_{i_1} > \alpha_{j(i_0)} + \sum\limits_{i' \in I_{j(i_0)}} \beta_{i'} - \beta_{i_0}$. Другими словами, плата за хранение партии i_1 в единицу времени больше штрафа за задержку заказа $j(i_0)$ и платы за хранение всех его партий, кроме последней. Тогда простой станка на ε перед началом партии i_1 приводит к задержке последнего заказа $T_{j(i_0)} = \varepsilon$, увеличению времени хранения его партий и падению целевой функции (1) на величину

$$\Delta F(\varepsilon) = \Big(\beta_{i_1} - \alpha_{j(i_0)} - \sum_{i' \in I_{j(i_0)}} \beta_{i'} + \beta_{i_0}\Big)\varepsilon.$$

Плата за хранение партии i_0 не меняется, так как величины $C_{j(i_0)}$ и c_{i_0} увеличиваются одновременно на ε и согласно (7), (8) величина e_{i_0} остаётся прежней. Следовательно, станок должен простаивать перед выполнением партии i_1 в течение времени $\tau = d_{j(i_1)} - c_{i_1}$, чтобы закончить партию i_1 в момент времени $d_{j(i_1)}$.

Рассмотрим общий случай. Для упрощения обозначений считаем, что перестановка σ имеет вид $\sigma(i) = n - i + 1$, $i \in I$. Будем двигаться по этой перестановке от конца к началу, отодвигая время старта партий в конце расписания, если это приводит к падению целевой функции. Предположим, что к началу шага l величины s_i для $i = 1, \ldots, l-1$ уже скорректированы наилучшим образом и требуется определить величину простоя станка перед выполнением партии l. Введём понятие блока, понимая под этим максимальную подпоследовательность партий, выполняемых без простоя станка [10].

Предположим, что партия l не принадлежит блоку, начинающемуся с партии $l-1$ (рис. 1), т. е. станок простаивал перед выполнением партии $l-1$. Назовём этот блок B_{l-1} по номеру первой партии в нём. Рассмотрим три случая.

Случай 1: $d_{j(l)} \leqslant c_l$. Условия (7) становятся неактивными, и величина e_l определяется равенством $e_l = C_{j(l)} - c_l$. Если партия l была последней в этом заказе, т. е. $C_{j(l)} = c_l$, то простой станка перед партией l только увеличит задержку $T_{j(l)}$ и приведёт к росту целевой функции.

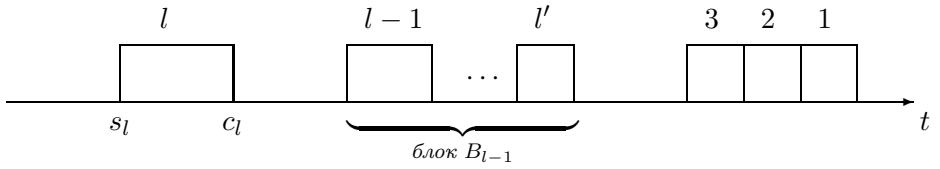

Рис. 1. Расписание перед шагом l

Если же эта партия не была последней в заказе, т. е. $C_{j(l)} > c_l$, то простой станка в течение $\tau = s_{l-1} - c_l$ сокращает плату за хранение партии, не меняет задержку заказа $T_{j(l)}$ и приводит к появлению нового блока B_l вместо блока B_{l-1}.

Случай 2: $c_l < d_{j(l)} \leqslant s_{l-1}$. Простой станка в течение $\tau = d_{j(l)} - c_l$ сокращает плату за хранение партии и не меняет задержку заказа. Дальнейший простой сводит задачу к предыдущему случаю.

Случай 3: $s_{l-1} < d_{j(l)}$. Простой станка в течение $\tau = s_{l-1} - c_l$ сокращает плату за хранение партии, не меняет задержку заказа и снова приводит к появлению нового блока B_l вместо блока B_{l-1}.

Рассмотрим ситуацию, когда партия l является первой в своём блоке, а после этого блока выполняются несколько других блоков. Небольшой простой на $\varepsilon > 0$ перед выполнением партии l влечёт за собой такой же сдвиг по времени для всех партий блока B_l. Аналогично предыдущему случаю можно вычислить изменение целевой функции для каждой партии блока и, пользуясь кусочно-линейностью функции, получить линейное приращение от ε. Заметим, что при положительной задержке заказа необходимо учитывать рост платы за хранение всех партий заказа, а не только партий данного блока. Если коэффициент этого приращения свидетельствует о падении целевой функции, то величина простоя полиномиально вычисляется по директивным срокам партий из блока B_l и текущим значениям переменных c_i, $i \leqslant l$, а также по времени старта следующего блока. Таким образом, на каждом шаге алгоритма величины s_i либо растут, либо не меняются, блоки могут появляться и исчезать, а целевая функция монотонно не возрастает. Общая схема алгоритма декодирования перестановки σ выглядит следующим образом.

Алгоритм декодирования

Шаг 1. Вычислить значения переменных $u_{ii'}$ по перестановке σ.

Шаг 2. Применить жадную эвристику для запасов на складе и определить значения переменных x_i, y_i, z_i, $i \in I$.

Шаг 3. Пользуясь условиями (2), (4), найти значения переменных s_i, $i \in I$, предполагая отсутствие простоев станка.

Шаг 4. Двигаясь с конца перестановки σ, скорректировать значения переменных s_i, $i \in I$, вычисляя оптимальную длительность простоя станка перед выполнением каждой партии.

Шаг 5. Определить значения остальных переменных по уже известным значениям s_i, $i \in I$, и вычислить значение целевой функции $F(\sigma)$.

Алгоритм декодирования полиномиален при любой полиномиальной эвристике использования запасов на складе. Действительно, для реализации шага 1 требуется $O(n^2)$ операций при заполнении матрицы $u_{ii'}$. Если на шаге 2 игнорировать трудоёмкость жадной эвристики, то потребуется $O(n)$ операций для вычисления значений переменных x_i, y_i, z_i, $i \in I$. На шаге 3 можно найти время старта всех партий с трудоёмкостью $O(n)$. Шаг 4 состоит из n итераций для вычисления простоя станка перед началом каждой партии. На каждой итерации вычисляется длина простоя с учётом всех партий заказа, если он опаздывал, и всех блоков, выполнявшихся после данного. Всего блоков не больше n. Если блок состоит из l партий, то приращение целевой функции меняется не более $l + 1$ раз в зависимости от того, склеиваются ли блоки или появляются новые запаздывающие заказы. В итоге требуется не более $O(n^2)$ операций для вычисления простоя перед очередной партией, и общая трудоёмкость шага 4 оценивается сверху величиной $O(n^3)$. Реализация шага 5 требует $O(n)$ операций.

Теорема 1. *Алгоритм декодирования строит на шаге* 4 *оптимальное расписание для заданных значений переменных* $x_i, y_i, z_i, u_{ii'}$, $i, i' \in I$.

Доказательство. Перед шагом 4 все булевы переменные задачи уже определены по перестановке σ, и решаемая оптимизационная задача по сути становится задачей линейного программирования. Следовательно, любой локальный минимум является глобальным. Так как время старта каждой партии однозначно определяет значения остальных переменных, достаточно проверить локальную оптимальность только этих переменных. Заметим, что в ходе работы алгоритма декодирования величины s_i, $i \in I$, не убывают с ростом числа шагов. На шаге l находится оптимальное время старта партии l, а дальнейший рост этой величины приводит только к росту целевой функции. Рост величин s_i, $i < l$, происходит только в том случае, если они оказываются в одном блоке с партией l и перед блоком выгоден простой станка. Значит, любое уменьшение величин s_i, $i < l$, также приводит к росту целевой функции, что и свидетельствует о получении локального оптимума.

Замечание 1. Кодировка решений в виде перестановок имеет недостатки, которые трудно исправить при жадном использовании склада.

В этой схеме перестановка не рассматривается целиком. Решения принимаются на каждом шаге отдельно. Рассмотрим пример из двух однотипных партий разных заказов с общим директивным сроком $d = 10$ и $p_1 = p_2 = 5$. Переналадка станка нулевая, $\beta = 1$, склад пустой. Заказы будут выполнены в срок: $T_1 = T_2 = 0$. При жадной схеме использования склада получим $s_1 = 0$, $c_1 = 5$, $s_2 = 5$, $c_2 = 10$, $y_1 = y_2 = 0$, $F(\sigma) = 5$. Оптимальное решение $s_1 = 0$, $c_1 = 10$, $s_2 = 10$, $c_2 = 10$, $y_1 = 0$, $y_2 = 1$, $F(\sigma) = 0$.

3.2. Общая схема локального поиска. Методы локального поиска и, в частности, метаэвристики проявили себя как мощное средство решения NP-трудных задач комбинаторной оптимизации [25]. Они успешно применялись для решения задач размещения [22], маршрутизации [13,14], кластеризации [3,20], двухуровневого программирования [4,5,16], балансировки нагрузки на серверы [1, 2] и др. Получены условия сходимости и границы возможностей таких методов [6, 9, 19, 21]. Ниже предлагается алгоритм локального поиска с запретами, который просматривает различные перестановки в поисках перестановки с наименьшим значением целевой функции [15].

Введём понятие окрестности для перестановки σ. Обозначим через $N(\sigma)$ множество перестановок, которые получаются из перестановки σ применением одной из двух операций: *замена* или *сдвиг*. В операции замена выбираются две партии и они меняются местами в перестановке. В операции сдвиг выбирается одна партия и она сдвигается на новое место в перестановке. Легко проверить, что из любой перестановки можно получить любую другую перестановку и, в частности, оптимальную перестановку, выполняя не более n переходов в соседнее решение по этой окрестности. В качестве рандомизированной окрестности $\mathcal{N}_p(\sigma)$ будем использовать случайное подмножество элементов из окрестности $N(\sigma)$. Каждый элемент окрестности $N(\sigma)$ включается в окрестность $\mathcal{N}_p(\sigma)$ с вероятностью p независимо от других элементов. В качестве списка запретов Tabu будем использовать пары элементов перестановки при их замене и элемент с его старым местом при сдвиге элемента на новое место. Длина списка запретов L будет постоянной величиной. Сам список содержит информацию только о последних модификациях перестановки. Общая схема локального поиска с запретами выглядит следующим образом.

Локальный поиск с запретами

Шаг 1. Выбрать случайную перестановку σ, параметры p и L, число итераций T и порог диверсификации T_d. Положить $F^* = F(\sigma)$, $t = 0$, Tabu $= \varnothing$.

Шаг 2. Пока $t \leqslant T$ повторять следующие шаги.

1) Сформировать окрестность $\mathcal{N}_p(\sigma)$.
2) Найти $F(\sigma') = \min\{F(\sigma_1) \mid \sigma_1 \in \mathcal{N}_p(\sigma) \setminus \text{Tabu}\}$.
3) Положить $\sigma \leftarrow \sigma'$, $t \leftarrow t + 1$ и обновить список запретов Tabu.
4) Если $F^* > F(\sigma)$, то положить $F^* \leftarrow F(\sigma)$, $\sigma^* \leftarrow \sigma$.
5) Если рекорд F^* не менялся в течение T_d итераций, то выбрать несколько запаздывающих заказов и сдвинуть их партии случайным образом на более ранние позиции в перестановке.

Шаг 3. Выдать наилучшее найденное решение σ^*.

Параметры алгоритма T, T_d, p и L подбираются под размерность решаемой задачи: чем больше партий, тем больше требуется итераций, меньше доля просматриваемой окрестности и длина списка запретов.

4. Численные эксперименты

Для проверки качества получаемых решений данный алгоритм был запрограммирован на языке C# и протестирован на примерах с числом заказов до 20, числом партий 60 при 45 различных типах продукции, а также на примерах реальной размерности месячного планирования. Параметры производства и склада взяты близкими к реальным и выбирались из следующих интервалов: $q_i \in [20, 100]$, $p_k \in [1, 5]$, $v_i \in [3, 400]$, $h_k \in [0, 200]$. При этом запасы на складе имеются только для 5% типов продукции. Матрица переналадок $\Delta_{kk'}$ содержит только три различных значения: 0, 30, 60, что соответствует либо переходу на ту же продукцию, но другого цвета, либо переходу на продукцию другого размера с заменой оснастки. Директивные сроки выбирались трёх типов: $d_j \in \{0', 480', 960'\}$, т. е. часть заказов уже запаздывала, часть требовалась к началу следующего дня, и остальные заказы должны быть выполнены через два дня. Штрафные коэффициенты α_j за опоздание заказов выбирались из множества $\{1, 5, 10, 15, 20\}$, что соответствовало разбиению заказов на пять приоритетных групп. Плата за хранение продукции на складе вычислялась по формуле $\beta_i = 0{,}001\gamma_i v_i$, где стоимость хранения единичной продукции γ_i принадлежала интервалу $[100, 400]$. Производственная программа планировалась на неделю, так как только на таких небольших примерах Gurobi удавалось получить достаточно качественные решения за час работы на рабочей станции, оснащённой двумя шестиядерными процессорами Intel Xeon X5675 3,07 ГГц и 32 Гб оперативной памяти. При месячном планировании Gurobi не удавалось получить даже допустимого решения линеаризованной задачи.

Разработанный алгоритм локального поиска с запретами предъявляет в качестве ответа наилучшее найденное решение, которое по сути является случайной величиной. Поэтому каждый пример решался 20 раз и вычислялись наименьшее, наибольшее и среднее получаемые значения

Таблица 1

Значения целевой функции при 5 заказах

	Gurobi	Среднее	Min	Max	%
1	62777,73	34825,48	28965,98	35533,16	44,53
2	33245,88	23939,42	23835,38	24843,23	27,99
3	51507,31	47358,39	46289,56	47641,95	8,06
4	33446,41	28951,65	28920,13	29021,49	13,44
5	21069,09	12013,54	12012,56	12015,14	42,98
6	63864,28	56878,49	56813,08	57288,02	10,94
7	75883,14	21432,76	19683,94	24349,88	71,76
8	83714,59	82621,13	74014,87	94893,85	1,31
9	22471,63	19035,85	16041,12	20919,94	15,29
10	52939,39	31245,45	30509,55	36402,27	40,98

целевой функции. В табл. 1–3 представлены результаты расчётов для 5, 10 и 15 заказов. В каждой таблице столбец Gurobi показывает значение целевой функции, полученное этим пакетом. Столбцы Среднее, Min, Max показывают соответственно среднее, минимальное и максимальное значения целевой функции, полученные алгоритмом локального поиска. Последний столбец показывает процент превышения значений для Gurobi над средним значением. В каждой таблице содержатся результаты расчётов для десяти примеров с разными по составу заказами из 60 партий. Параметры алгоритма $T = 40000$, $T_d = 450$, $p = 0{,}5$, $L = 100$ выбирались так, чтобы время работы было ограничено 10 минутами. При решении задач месячного планирования достаточно получаса расчётов, но параметр рандомизации окрестности можно уменьшить до $p = 0{,}1$ и сократить длину списка запретов вдвое.

Таблица 2

Значения целевой функции при 10 заказах

	Gurobi	Среднее	Min	Max	%
1	27665,64	12883,01	12274,93	14527,88	53,43
2	41255,06	26489,11	26014,47	26832,46	35,79
3	16730,79	11403,88	11371,79	11687,72	31,84
4	43611,20	39310,70	39262,01	39364,72	9,86
5	40320,10	21662,61	20844,48	22246,22	46,27
6	23489,95	14572,12	14103,39	14911,40	37,96
7	31795,67	10666,53	10422,23	10822,02	66,45
8	27610,27	11801,30	11742,01	12012,91	57,26
9	34069,08	22437,38	21824,16	22753,48	34,14
10	35030,59	23157,11	23103,16	23366,32	33,89

Таблица 3

Значения целевой функции при 15 заказах

	Gurobi	Среднее	Min	Max	%
1	24481,86	11550,48	11435,35	11791,55	52,82
2	22378,51	10800,27	10762,96	10919,97	51,74
3	26477,23	19221,23	18960,90	19250,23	27,40
4	22095,64	14433,47	14431,83	14437,40	34,68
5	37684,67	9720,78	9667,72	9760,53	74,20
6	39337,26	37163,83	37071,67	37981,49	5,53
7	78120,99	49521,75	49366,87	49677,20	36,61
8	43845,56	33033,50	32935,72	33253,46	24,66
9	31981,38	9537,24	9481,90	9701,74	70,18
10	57244,71	50235,51	49541,55	50707,99	12,24

5. Заключение

В работе исследовалась новая задача построения расписаний для одного станка. В отличие от предшествующих задач здесь присутствует склад готовой продукции с разнородным начальным запасом. Вводится понятие минимально допустимого объёма производства, что вынуждает производить больше продукции, чем необходимо клиентам, и порождает новые запасы на складе. Учёт этих технологических особенностей приводит к задаче смешанного частично-целочисленного программирования с нелинейными ограничениями. Введением вспомогательных переменных эти ограничения удаётся заменить линейными с небольшим ростом размерности задачи. Такой подход открывает возможности для получения точного решения классическими методами. К сожалению, точное решение удаётся получить только на примерах малой размерности, $n \leqslant 30$, что явно недостаточно для практики.

В связи с этим в работе разработан приближённый алгоритм локального поиска с запретами, который в качестве кодировки решений использует перестановки партий продукции. Для распределения складских запасов применяется жадная эвристика, взятая из практики. Разработан алгоритм декодирования перестановки, который находит оптимальное расписание работы станка после распределения складских запасов. Проведены численные эксперименты на случайных тестовых примерах, при формировании которых использовалась реальная информация о длительностях переналадки станка, минимальных партиях и времени производства разных типов продукции. Полученные результаты свидетельствуют о высокой эффективности разработанного алгоритма и возможности быстро решать задачи месячного планирования.

В качестве направлений дальнейших исследований интересно получить алгоритмы решения задачи для нескольких параллельных станков. Каждый из них может иметь свой склад и свою себестоимость продукции. Кроме того, продукция может поставляться в магазины и транспортные затраты тоже могут быть различными для разных станков. Оптимизация таких производственных цепей поставок представляется авторам чрезвычайно интересной задачей.

ЛИТЕРАТУРА

1. **Давыдов И. А., Кононова П. А., Кочетов Ю. А.** Локальный поиск с окрестностью экспоненциальной мощности для задачи балансировки нагрузки на серверы // Дискрет. анализ и исслед. операций. 2014. Т. 21, № 6. С. 21–34.
2. **Давыдов И. А., Мельников А. А., Кононова П. А.** Локальный поиск для задач балансировки нагрузки серверов большой размерности // Автоматика и телемеханика. 2017. № 3. С. 34–50.
3. **Кочетов Ю. А., Панин А. А., Плясунов А. В.** Генетический локальный поиск и сложность аппроксимации задачи балансировки нагрузки на серверы // Автоматика и телемеханика. 2017. № 3. С. 51–62.
4. **Лавлинский С. М., Панин А. А., Плясунов А. В.** Двухуровневая модель планирования государственно-частного партнерства // Автоматика и телемеханика. 2015. № 11. С. 89–103.
5. **Панин А. А., Пащенко М. Г., Плясунов А. В.** Двухуровневые модели конкурентного размещения производства и ценообразования // Автоматика и телемеханика. 2014. № 4. С. 153–169.
6. **Aarts E. H. L., Korst J. H. M., van Laarhoven P. J. M.** Simulated annealing // Local search in combinatorial optimization. Chichester: Wiley, 1997. P. 91–120.
7. **Allahverdi A.** The third comprehensive survey on scheduling problems with setup times/costs // Eur. J. Oper. Res. 2015. Vol. 246, No. 2. P. 345–378.
8. **Bigras L.-P., Gamache M., Savard G.** The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times // Discrete Optim. 2008. Vol. 5, No. 4. P. 685–699.
9. **Brimberg J., Hansen P., Mladenović N.** Attraction probabilities in variable neighborhood search // 4OR. 2010. Vol. 8, No. 2. P. 181–194.
10. **Brucker P.** Scheduling algorithms. Berlin; Heidelberg: Springer, 2007.
11. **Chen B., Potts C. N., Woeginger G. J.** A review of machine scheduling: Complexity, algorithms and approximability // Handbook of combinatorial optimization. Vol. 3. Dordrecht: Kluwer Acad. Publ., 1998. P. 21–169.
12. **Dolgui A., Eremeev A. V., Kovalyov M. Y., Kuznetsov P. M.** Multi-product lot sizing and scheduling on unrelated parallel machines // IIE Trans. 2010. Vol. 42, No. 7. P. 514–524.

13. Erzin A. I., Mladenović N., Plotnikov R. V. Variable neighborhood search variants for min-power symmetric connectivity problem // Comput. Oper. Res. 2017. Vol. 78. P. 557–563.
14. Erzin A. I., Plotnikov R. V. Using VNS for the optimal synthesis of the communication tree in wireless sensor networks // Electron. Notes Discret. Math. 2015. Vol. 47. P. 21–28.
15. Glover F., Laguna M. Tabu Search. Norwell, MA: Kluwer Acad. Publ., 1997.
16. Iellamo S., Alekseeva E. V., Chen L., Coupechoux M., Kochetov Yu. A. Competitive location in cognitive radio networks // 4OR. 2015. Vol. 13, No. 1. P. 81–110.
17. Jackson J. R. Scheduling a production line to minimize maximum tardiness. Res. Rep. No. 43. Los Angeles: Univ. Calif., 1955.
18. Janak S. L., Floudas Ch. A., Kallrath J., Vormbrock N. Production scheduling of a large-scale industrial batch plant. I. Short-term and medium-term scheduling // Ind. Eng. Chem. Res. 2006. Vol. 45. P. 8234–8252.
19. Kochetov Yu. A. Facility location: discrete models and local search methods // Combinatorial optimization. Methods and applications. Amsterdam: IOS Press, 2011. P. 97–134.
20. Kochetov Yu., Kondakov A. A. VNS matheuristic for a bin packing problem with a color constraint // Electron. Notes Discret. Math. 2017. Vol. 58. P. 39–46.
21. Korte B. H., Vygen J. Combinatorial optimization: Theory and algorithms. New York: Springer, 2012.
22. Mladenović N., Brimberg J., Hansen P., Moreno-Pérez J. A. The p-median problem: A survey of metaheuristic approaches // Eur. J. Oper. Res. 2007. Vol. 179, No. 3. P. 927–939.
23. Pochet Y., Wolsey L. A. Production planning by mixed integer programming. New York: Springer, 2005.
24. Smith W. E. Various optimizers for single-stage production // Nav. Res. Logist. Q. 1956. Vol. 3, No. 1–2. P. 59–66.
25. Talbi E.-G. Metaheuristics: From design to implementation. Hoboken, NJ: Wiley, 2009.

Кононова Полина Александровна
Кочетов Юрий Андреевич

Статья поступила
11 октября 2018 г.
После доработки —
4 февраля 2019 г.
Принята к публикации
27 февраля 2019 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
April–June 2019. Vol. 26, No. 2. P. 60–78

UDC 519.8 DOI 10.33048/daio.2019.26.634

A LOCAL SEARCH ALGORITHM FOR THE SINGLE MACHINE SCHEDULING PROBLEM WITH SETUPS AND A STORAGE

P. A. Kononova[1,2,a] and *Yu. A. Kochetov*[1,2,b]

[1]Sobolev Institute of Mathematics,
4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia

[2]Novosibirsk State University,
1 Pirogov Street, 630090 Novosibirsk, Russia

E-mail: [a]pkononova@math.nsc.ru, [b]jkochet@math.nsc.ru

Abstract. We present a new mathematical model for a single machine scheduling problem originated from the tile industry. The model takes into account the sequence-dependent setup times, the minimal batch size, heterogeneous orders of customers, and a stock in storage. As the objective function we use the penalty for tardiness of the customers' orders and the total storage cost for final products. A mixed-integer linear programming model is applied for small test instances. For real-world applications, we design a randomized tabu search algorithm. The computational results for some test instances from a Novorossiysk company are discussed. Tab. 3, illustr. 1, bibliogr. 25.

Keywords: tabu search, scheduling, due date, tardiness, setup time.

REFERENCES

1. **I. A. Davydov, P. A. Kononova,** and **Yu. A. Kochetov,** Local search with an exponential neighborhood for the servers load balancing problem, *Diskretn. Anal. Issled. Oper.,* **21**, No. 6, 21–34, 2014 [Russian]. Translated in *J. Appl. Ind. Math.,* **9**, No. 1, 27–35, 2015.
2. **I. A. Davydov, A. A. Melnikov,** and **P. A. Kononova,** Local search for load balancing problems for servers with large dimension, *Avtom. Telemekh.,* No. 3, 34–50, 2017 [Russian]. Translated in *Autom. Remote Control,* **78**, No. 3, 412–424, 2017.
3. **Yu. A. Kochetov, A. A. Panin,** and **A. V. Plyasunov,** Genetic local search and hardness of approximation for the server load balancing problem, *Avtom. Telemekh.,* No. 3, 51–62, 2017 [Russian]. Translated in *Autom. Remote Control,* **78**, No. 3, 425–434, 2017.

4. **S. M. Lavlinskii, A. A. Panin,** and **A. V. Plyasunov,** A bilevel planning model for public-private partnership, *Avtom. Telemekh.,* No. 11, 89–103, 2015 [Russian]. Translated in *Autom. Remote Control,* **76**, No. 11, 1976–1987, 2015.
5. **A. A. Panin, M. G. Pashchenko,** and **A. V. Plyasunov,** Bilevel competitive facility location and pricing problems, *Avtom. Telemekh.,* No. 4, 153–169, 2014 [Russian]. Translated in *Autom. Remote Control,* **75**, No. 4, 715–727, 2014.
6. **E. H. L. Aarts, J. H. M. Korst,** and **P. J. M. van Laarhoven,** Simulated annealing, in *Local Search in Combinatorial Optimization,* pp. 91–120, Wiley, Chichester, 1997.
7. **A. Allahverdi,** The third comprehensive survey on scheduling problems with setup times/costs, *Eur. J. Oper. Res.,* **246**, No. 2, 345–378, 2015.
8. **L.-P. Bigras, M. Gamache,** and **G. Savard,** The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times, *Discrete Optim.,* **5**, No. 4, 685–699, 2008.
9. **J. Brimberg, P. Hansen,** and **N. Mladenović,** Attraction probabilities in variable neighborhood search, *4OR,* **8**, No. 2, 181–194, 2010.
10. **P. Brucker,** *Scheduling Algorithms,* Springer, Heidelberg, 2007.
11. **B. Chen, C. N. Potts,** and **G. J. Woeginger,** A review of machine scheduling: complexity, algorithms and approximability, in *Handbook of Combinatorial Optimization,* Vol. 3, pp. 21–169, Kluwer Acad. Publ., Dordrecht, 1998.
12. **A. Dolgui, A. V. Eremeev, M. Ya. Kovalyov,** and **P. M. Kuznetsov,** Multi-product lot sizing and scheduling on unrelated parallel machines, *IIE Trans.,* **42**, No. 7, 514–524, 2010.
13. **A. I. Erzin, N. Mladenović,** and **R. V. Plotnikov,** Variable neighborhood search variants for min-power symmetric connectivity problem, *Comput. Oper. Res.,* **78**, 557–563, 2017.
14. **A. I. Erzin** and **R. V. Plotnikov,** Using VNS for the optimal synthesis of the communication tree in wireless sensor networks, *Electron. Notes Discrete Math.,* **47**, 21–28, 2015.
15. **F. Glover** and **M. Laguna,** *Tabu Search,* Norwell, MA: Kluwer Acad. Publ., 1997.
16. **S. Iellamo, E. V. Alekseeva, L. Chen, M. Coupechoux,** and **Yu. A. Kochetov,** Competitive location in cognitive radio networks, *4OR,* **13**, No. 1, 81–110, 2015.
17. **J. R. Jackson,** Scheduling a production line to minimize maximum tardiness, *Res. Rep.,* No. 43, Univ. Calif., Los Angeles, 1955.
18. **S. L. Janak, Ch. A. Floudas, J. Kallrath,** and **N. Vormbrock,** Production scheduling of a large-scale industrial batch plant. I. Short-term and medium-term scheduling, *Ind. Eng. Chem. Res.,* **45**, 8234–8252, 2006.
19. **Yu. A. Kochetov,** Facility location: Discrete models and local search methods, *Combinatorial Optimization: Methods and Applications,* pp. 97–134, IOS Press, Amsterdam, 2011.
20. **Yu. A. Kochetov** and **A. A. Kondakov,** VNS matheuristic for a bin packing problem with a color constraint, *Electron. Notes Discrete Math.,* **58**, 39–46, 2017.

21. B. Korte and **J. Vygen,** *Combinatorial Optimization: Theory and Algorithms,* Springer, Heidelberg, 2012 (Algorithms Comb., Vol. 21).

22. N. Mladenović, J. Brimberg, P. Hansen, and **J. A. Moreno-Pérez,** The p-median problem: A survey of metaheuristic approaches, *Eur. J. Oper. Res.,* **179**, No. 3, 927–939, 2007.

23. Y. Pochet and **L. A. Wolsey,** *Production Planning by Mixed Integer Programming,* Springer, New York, 2006.

24. W. E. Smith, Various optimizers for single-stage production, *Nav. Res. Logist. Q.,* **3**, No. 1–2, 59–66, 1956.

25. E.-G. Talbi, *Metaheuristics: From Design to Implementation,* Wiley, Hoboken, NJ, 2009.

Polina A. Kononova
Yury A. Kochetov

Received October 11, 2018
Revised February 4, 2019
Accepted February 27, 2019