Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования

УРАЛЬСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ
имени первого Президента России Б. Н. Ельцина

Институт математики и компьютерных наук
Кафедра алгебры и дискретной математики

Задача построения графика горячей прокатки, сбалансированного по видам продукции.
Алгоритмы построения графика.

Допущен к защите

____

«__»_________ 2016 г.

Квалификационная работа
на степень магистра наук
по направлению «Математика
и компьютерные науки»
студента группы МГКН-2
Березина Антона Александровича

Научный руководитель
Баранский Виталий Анатольевич,
доктор физико-математических
наук, профессор

Екатеринбург
2016

Эта страница появилась здесь, потому что я не умею нормально нумеровать страницы в TeX.

РЕФЕРАТ

Березин А.А. ЗАДАЧА ПОСТРОЕНИЯ ГРАФИКА ГОРЯЧЕЙ ПРОКАТКИ, СБАЛАНСИРОВАННОГО ПО ВИДАМ ПРОДУКЦИИ. РАЗРАБОТКА МОДЕЛИ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ,

выпускная квалификационная работа на степень магистра наук: стр. ?, рис. ?, табл. ?, форм. ?, библ. ? назв.

Ключевые слова: ОРИЕНТИРОВАННЫЕ ГРАФЫ, ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ, ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ, ГОРЯЧАЯ ПРОКАТКА

Объект исследования — раскрашенные вершинно-взвешенные ориентированные графы специального вида.

Цель работы — поиск в этих графах простых цепей, в которых суммарный вес узлов каждого цвета максимально близок к заданному значению.

Решение задачи поиска «сбалансированной» простой цепи сводится к решению серии задач целочисленного линейного программирования. Этот подход основан на результатах Леоновой С.И. по изучению структуры рассматриваемых графов [1] и позволяет существенно снизить размерность получаемых задач линейного программирования по сравнению с прямым использовании классических техник для их построения (например, [4]). Полученные результаты применяются для решения более общих задач в области планирования металлургического производства.

Оглавление

Глава 1

Введение

Работа посвящена построению и исследованию алгоритмов формирования графиков прокатки для непрерывных станов горячей прокатки. Задача построения графика прокатки состоит в том, чтобы из заданного множества партий слябов (производственных заготовок) выбрать подмножество заданного веса и сформировать из него последовательность, называемую графиком прокатки, удовлетворяющую заданным технологическим ограничениям, которые определяют допустимый порядок следования партий.

В задаче построения сбалансированного графика прокатки требуется оптимизировать график таким образом, чтобы суммарный вес слябов, относящихся к определенным типам продукции, был максимально близок к заданным значениям для каждого из этих типов.

Для решения данной задачи рассматривается раскрашенный вершинно-взвешенный ориентированный граф предшествования партий. Множеством узлов этого орграфа является множество партий, множество дуг формируется на основе технологических ограничений. Партия p связана дугой с партией q в том и только в том случае, когда партия q может непосредственно следовать за партией p в графике прокатки. Вес каждого узла равен суммарному весу слябов соответствующей партии, цвет узла определяется типом продукции. Задача построения сбалансированного графика прокатки формализуется как задача поиска в этом орграфе простой цепи, в которой суммарный вес узлов каждого цвета максимально близок к заданному значению.

Решение задачи поиска «сбалансированной» простой цепи сводится к решению серии задач вещественно-целочисленного линейного программирования. Этот подход основан на результатах авторов по изучению структуры графа предшествования партий и позволяет существенно снизить размерность получаемых задач линейного программирования по сравнению с прямым использовании классических техник для их построения (например, [4]).

В докладе будут приведены результаты вычислительных экспериментов с использованием реальных и смоделированных данных в системе IBM ILOG CPLEX.

Глава 2

Постановка задачи

Обозначим через P множество партий $\{p_1, p_2, \ldots, p_k\}$, где $k \in \mathbb{N}$. Обозначим через F множество видов продукции партий $\{f_1, f_2, \ldots, f_s\}$, где $s \in \mathbb{N}$.
Пусть

- $f : P \to F$ — функция, обозначающая вид продукции каждой партии;
- $\mu : F \to \mathbb{R}^+$ — функция, задающая желаемое распределение различных видов продукции в графике прокатки.
- $w : P \to \mathbb{R}^+$ — ширина партии;
- $t : P \to \mathbb{R}^+$ — толщина партии;
- $l : P \to \mathbb{R}^+$ — весовая функция (длина партии).
- $r : \mathbb{R}^+ \to \mathbb{R}^+$ — монотонно неубывающая функция, определяющая максимальную величину допустимой разности значений толщины между двумя соседними партиями;
- $\mathbb{W} \in \mathbb{R}^+$ — величина, определяющая максимальную величину допустимой разности значений ширины между двумя соседними партиями.

Геометрические параметры партий являются основным источником накладываемых ограничений на порядок следования партий друг за другом в графике прокатки. Партия q может непосредственно следовать за партией p в графике прокатки в том и только в том случае, когда выполнены:

1. ограничение «для перехода по толщине»:
$$|t_p - t_q| \leq \min\{r(t_p), r(t_q)\} \tag{2.1}$$
2. ограничение «для перехода по ширине»:
$$0 \leq w_p - w_q \leq \mathbb{W} \tag{2.2}$$

Пусть G ориентированный граф на множестве узлов P с множеством дуг $E \subseteq P \times P$ таким, что $pq \in E$ в том и только в том случае, когда партии p и q различны, и q может непосредственно следовать за p в графике прокатки.

Множество всех допустимых графиков прокатки совпадает со множеством $\mathbb{B}(G)$ всех простых цепей в G.

Для простой цепи z через $l(z)$ обозначим её вес:

$$l(z) \triangleq \sum_{p \in P(z)} l(p) \tag{2.3}$$

Сформулируем **задачу** поиска допустимого решения — найти в $\mathbb{B}(G)$ такую цепь, что выполнены ограничения:

$$L_{min} \leq \sum_{p \in P(z)} l(p) \leq L_{max} \tag{2.4}$$

$$\forall k \in \overline{1,s} \quad \mu_k - \delta_k \leq \sum_{p \in P(z), f(p)=f_k} l(p) \leq \mu_k + \delta_k \tag{2.5}$$

где δ_k — заданная величина допустимого отклонения от целевого значения для каждого вида продукции.

Также целесообразно рассматривать оптимизационную задачу: найти в $\mathbb{B}$ простую цепь z, такую что:

$$\sum_{i=1}^{s} \left| \mu(f_i) - \frac{\sum\limits_{f(p_j)=f_i, p_j \in z} l(p_j)}{\sum\limits_{p_j \in z} l(p_j)} \right| \to \min_{p_j \in z} \tag{2.6}$$

Пусть p - произвольный узел в G. Обозначим через:

- $In(p)$ — множество узлов q таких, что дуга $qp \in E$
  $|In(p)|$ — входящая валентность узла p
- $In^+(p) = \{q \in In(p) | w_q > w_p\}$
- $Out(p)$ — множество узлов q таких, что дуга $pq \in E$
  $|Out(p)|$ — исходящая валентность узла p
- $Succ(p)$ — множество всех узлов q, для которых существует ориентированный путь pq в графе G. Из общей теории графов известно, что в этом случае также существует и простая цепь из p в q.
- $Pred(p)$ — множество всех узлов q, для которых существует ориентированный путь из q в p в графе G. Отметим, что в этом случае также существует и простая цепь из q в p.
- $W = \{w_p | p \in P\}$ — всевозможные значения ширин узлов
- $P_w = \{p | w_p = w\}$, где $w \in W$, — множество узлов, имеющих данную ширину w.
- G_w — граф, индуцированный множеством P_w.

Глава 3

Метод решения

Для эффективного решения задачи предложен следующий подход:

1. Генерируется некоторое начальное приближение. Для этого используется алгоритм построения простой цепи максимального веса ([3]) в графе G. Можно использовать и другие подходы.

2. Решается задача локальной оптимизации найденного приближения методами математического программирования.

В ходе работы было разработано несколько моделей для решения задачи локальной оптимизации.

Глава 4

Математическая модель

P — множество партий, $G_{w_1}, G_{w_2}, \ldots, G_{w_m}$ — набор подграфов графа следования партий, которые далее будем называть «слоями».

Пронумеруем партии следующим образом : p_{ij} — j-я партия в i-м слое.

Каждой партии p_{ij} соответствует переменная

$b_{ij} = \begin{cases} 1, & \text{если } p_{ij} \text{ вошла в график прокатки;} \\ 0, & \text{если } p_{ij} \text{ не вошла в график прокатки.} \end{cases}$

Обозначим

- $n_i = |P(G_{w_i})|$ — количество партий в слое G_{w_i}.
- $l_{ij} = l(p_{ij})$ — вес p_{ij} партии.
- L_{min}, L_{max} — заданные минимальное и максимальное значения суммарной длины партий в графике.
- $f_1, f_2, \ldots, f_s$ — набор видов продукции.
- $\mu_i = \mu(f_i) = (\underline{\mu}_i, \overline{\mu}_i)$ — распределение видов продукции, заданное диапазоном суммарной длины партий, имеющих f_i вид продукции.
- $f_{ij} = f(p_{ij})$ — вид продукции p_{ij} партии.

Ограничение на суммарный вес

$$L_{min} \leq \sum_{i=1}^{m} \sum_{j=1}^{l_i} b_{ij} l_{ij} \leq L_{max} \tag{4.1}$$

Ограничение на виды продукции

$$\forall k \in \overline{1, s} \quad \underline{\mu}_k \leq \sum_{f_{ij}=f_k} b_{ij} l_{ij} \leq \overline{\mu}_k \tag{4.2}$$

Эти ограничения являются базовыми и далее меняться не будут. Подход к ограничениям предшествования может быть различным, и в зависимости от выбранного подхода можно получать более быстрые либо более качественные решения. Далее рассмотрим несколько подходов к формализации ограничений предшествования партий в графике прокатки.

4.1 Первая модель

В рамках этой модели рассматривается набор подграфов G_{w_i}, через которые проходит простая цепь, построенная на стадии генерации начального приближения. Фиксируются граничные узлы, через которые эти подграфы соединяются. Специальными ограничениями накладывается требование на то, чтобы эти узлы входили в построенное решение, таким образом удается избавиться от необходимости обеспечивать связность между соседними слоями в выбранном наборе подграфов. Необходимо лишь обеспечить связность решения в рамках отдельных подграфов.

Ограничение предшествования

- G_{w_i} — слой, s_i, e_i — заданная пара узлов (вход и выход) для слоя G_{w_i}.
- Пусть $p_{ij_0} \in G_{w_i}$, причем $p_{ij_0} \neq s_i$ и $p_{ij_0} \neq e_i$.

Пусть $p \in G_{w_i}$. Тогда $\forall q \in G_{w_i} :\ pq \in E$ обозначим

- $\overrightarrow{Out}_{G_{w_i}}(p) = \{q \mid t_q > t_p\}$, $\overrightarrow{J_p} = \{j | p_{ij} \in \overrightarrow{Out}_{G_{w_i}}(p_{ij_0})\}$.
- $\overleftarrow{Out}_{G_{w_i}}(p) = \{q \mid t_q < t_p\}$, $\overleftarrow{J_p} = \{j | p_{ij} \in \overleftarrow{Out}_{G_{w_i}}(p_{ij_0})\}$.

Тогда если $t(s_i) < t_{ij_0} < t(e_i)$, то

$$b_{ij_0} \leq \sum_{j' \in \overrightarrow{J_p}} b_{ij'} \quad b_{ij_0} \leq \sum_{j' \in \overleftarrow{J_p}} b_{ij'} \tag{4.3}$$

Если $t_{ij_0} < t(s_i)$, то

$$2b_{ij_0} \leq \sum_{j' \in \overrightarrow{J_p}} b_{ij'} \tag{4.4}$$

Если $t(e_i) < t_{ij_0}$, то

$$2b_{ij_0} \leq \sum_{j' \in \overleftarrow{J_p}} b_{ij'} \tag{4.5}$$

Локальная задача существования

Локальная задача существования состоит в том, чтобы найти решение следующей системы уравнений и неравенств, либо установить, что его не существует.

$$
\begin{cases}
L_{min} \le \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n_i} b_{ij} l_{ij} \le L_{max}; \\
\forall i \in \overline{1,m}: \ b_i^s = 1, \ b_i^e = 1; \\
\forall i \in \overline{1,m} \ \forall j_0 \in \overline{1,n_i} (p_{ij_0} \notin \{s_i, e_i\}): \\
\begin{cases}
b_{ij_0} \le \sum\limits_{j' \in \overrightarrow{J_p}} b_{ij'}, \quad b_{ij_0} \le \sum\limits_{j' \in \overleftarrow{J_p}} b_{ij'}, & \text{если } t(s_i) < t_{ij_0} < t(e_i); \\
2b_{ij_0} \le \sum\limits_{j' \in \overrightarrow{J_p}} b_{ij'}, & \text{если } t_{ij_0} < t(s_i); \\
2b_{ij_0} \le \sum\limits_{j' \in \overleftarrow{J_p}} b_{ij'}, & \text{если } t(e_i) < t_{ij_0}.
\end{cases} \\
\forall k \in \overline{1,s} \quad \underline{\mu_k} \le \sum\limits_{f_{ij} = f_k} b_{ij} l_{ij} \le \overline{\mu_k}.
\end{cases}
$$

Вложенная подсистема неравенств эквивалентна тому, что для партий, вошедших в решение, на подмножестве партий (узлов), принадлежащих одному блоку вместе с входом и выходом возможно построить путь в G с началом во входном узле и окончанием в выходном узле. Возможность собрать все партии в решении локальной задачи в один путь тогда вытекает из того, что входы и выходы в следующих друг за другом блоках выбраны согласованно.

Локальная задача существования может быть использована для получения решений и в более компактной схеме. В предыдущей работе [3] предложен алгоритм кубической временной трудоемкости, решающий задачу поиска пути максимального веса. По такому пути естественным образом строится набор блоков с входами и выходами. Используя эти данные в качестве начальных в локальной задаче существования иногда возможно построить решение. Вычислительные эксперименты, проведенные с использование системы IBM ILOG CPLEX, показали, что зачастую таким коротким путём возможно получить решение локальной задачи существования и, соответственно, решение исходной задачи. При этом решение получается за короткое время при числе партий порядка 100. При этом оптимальное решение локальной задачи зачастую строится за 1-2 секунды.

4.2 Вторая модель

Расширенная локальная задача существования

Пусть заданы :

- множество партий P,
- фиксированная последовательность подграфов $G_{w_1}, G_{w_2}, \dots, G_{w_m}$,
- ограничения(на вес, на предшествование, на виды продукции).

В рамках этой модели так же рассматривается фиксированный набор подграфов G_{w_i} полученный на стадии генерации начального приближения. Отличие от предыдущей задачи в том, что теперь не зафиксировано то, какие узлы могут быть входами и выходами, а значит требуется ввести ограничения, описывающие связность между соседними уровнями.

К модели добавляются следующие переменные:

Каждой партии p_{ij} соответствуют

$$s_{ij} = \begin{cases} 1, & \text{если } p_{ij} \text{ является входом в } i \text{ слой;} \\ 0, & \text{если } p_{ij} \text{ не является входом в } i \text{ слой.} \end{cases}$$

$$e_{ij} = \begin{cases} 1, & \text{если } p_{ij} \text{ является выходом из } i \text{ слоя;} \\ 0, & \text{если } p_{ij} \text{ не является выходом из } i \text{ слоя.} \end{cases}$$

А так же к существующим добавляются ограничения:

В каждом слое должен быть ровно один вход и ровно один выход

$$\forall i \ \sum_j s_{ij} = 1$$

$$\forall i \ \sum_j e_{ij} = 1$$

Входы и выходы должны быть включены в график прокатки

$$\forall i \ \forall j \ b_{ij} \geq s_{ij}$$

$$\forall i \ \forall j \ b_{ij} \geq e_{ij}$$

Определим множество узлов в k-том слое, которые были бы связаны с узлом, имеющим ширину t

$$O_k(t) = \{v \in B_k| \ |t(v) - t| \leq min\{r(t(v)), r(t)\}\}$$

И множество индексов этих узлов

$$J_k(t) = \{j| \ p_{kj} \in O_k(t)\}$$

Смысл следующего ограничения в том, что для каждого входа в слой, среди узлов-предков этого входа должен существовать узел, являющийся выходом

$$\forall i = 2, ..., m \ \forall j_0 \ s_{ij_0} \leq \sum_{j \in O_{i-1}(t_{ij})} e_{i-1j}$$

Соответствующее ограничение для выходов

$$\forall i = 1, ..., m-1 \ \forall j_0 \ e_{ij_0} \leq \sum_{j \in O_{i+1}(t_{ij})} s_{i+1j}$$

Ограничения предшествования в рамках отдельных подграфов заменяются двумя новыми ограничениями:

- $$b_{ij0} \leq \frac{1}{2} \sum_{j' \in \overrightarrow{J^o_{(b_{ij_0})}}} b_{ij'} + (1 - \frac{1}{2} \sum_{j \in \overrightarrow{J_{(b_{ij_0})}}} (s_{ij} + e_{ij}))$$

- $$b_{ij0} \leq \frac{1}{2} \sum_{j' \in \overleftarrow{J^o_{(b_{ij_0})}}} b_{ij'} + (1 - \frac{1}{2} \sum_{j \in \overleftarrow{J_{(b_{ij_0})}}} (s_{ij} + e_{ij}))$$

Где

$$\overrightarrow{Out}^o_{B_i}(p) = \{q \in B_i \mid t_q > t_p, \ \ |t_q - t_p| \leq min\{r(t_q), r(t_p)\}\}$$
$$\overrightarrow{J^o_p} = \{j|p_{ij} \in \overrightarrow{Out}^o_{B_i}(p_{ij_0})\}$$

и соответственно

$$\overleftarrow{Out}^o_{B_i}(p) = \{q \in B_i \mid t_q < t_p, \ \ |t_q - t_p| \leq min\{r(t_q), r(t_p)\}\}$$
$$\overleftarrow{J^o_p} = \{j|p_{ij} \in \overleftarrow{Out}^o_{B_i}(p_{ij_0})\}$$

4.3 Третья модель

Эта модель является развитием предыдущей задачи. В предыдущей задаче была задана фиксированная последовательность подграфов, в том смысле что для построения решения необходимо было, чтобы в него были включены узлы из каждого подграфа. Теперь же сформулируем ограничения для связности подграфов таким образом, чтобы избежать этой необходимости. Это позволит рассматривать в рамках третьей модели произвольные последовательности подграфов (а значит, мы сможем перейти от локальных задач существования к глобальной).

Для каждого подграфа G_{w_i} поставим в соответствие переменную

$$q_i = \begin{cases} 1, & \text{если подграф } G_{w_i} \text{ включен в решение (хотя бы 1 узел из } G_{w_i}\text{);} \\ 0, & \text{иначе.} \end{cases}$$

Если подграф G_{w_i} не включен, то ни один его узел не представлен в решении.

$$\forall i\ \forall j\ b_{ij} \le q_i$$

Если хотя бы один узел подграфа G_{w_i} включен, то подграф включен в решение.

$$\forall i\ q_i \le \sum_j b_{ij}$$

Если есть вход и есть выход, то подграф включен в решение.

$$\forall i\ q_i = \sum_j e_{ij} = \sum_j s_{ij}$$

Входы и выходы включены в решение.

$$\forall i\ \forall j\ b_{ij} \ge s_{ij}$$

$$\forall i\ \forall j\ b_{ij} \ge e_{ij}$$

Если подграф включен в решение, то среди узлов-предков входа в этот уровень должны быть включенные в решение узлы.

$$s_{ij} \le \sum_{k<l<i} q_l + \sum_{z \in In_{B_k}(p_{ij})} e_{kz} + (1 - q_k)$$

Аналогично для выходов.

$$e_{ij} \le \sum_{i<l<k} q_l + \sum_{z \in Out_{B_k}(p_{ij})} s_{kz} + (1 - q_k)$$

Плюс ограничения на связность в рамках подграфов не меняются, они такие же как и в предыдущей модели.

- $$b_{ij0} \le \frac{1}{2} \sum_{j' \in \overrightarrow{J^o_{(b_{ij0})}}} b_{ij'} + (1 - \frac{1}{2} \sum_{j \in \overrightarrow{J_{(b_{ij0})}}} (s_{ij} + e_{ij}))$$

- $$b_{ij0} \le \frac{1}{2} \sum_{j' \in \overleftarrow{J^o_{(b_{ij0})}}} b_{ij'} + (1 - \frac{1}{2} \sum_{j \in \overleftarrow{J_{(b_{ij0})}}} (s_{ij} + e_{ij}))$$

Глава 5

Программная реализация

В работе использован язык программирования Java.

5.1 Описание кода программы

5.1.1 Типы данных

5.1.2 Методы

5.2 Результаты экспериментов

Данные для тестов берутся из корпоративной сети ОАО "ММК". Обычно в распоряжении имеются выборки порядка 200-300 партий.

Реальные данные таковы, что обычно при больших выборках от 75% процентов узлов попадают в построенную цепь. Это хороший с практической точки зрения результат, т.к. на производстве такие большие графики прокатки избыточны и построенную цепь можно дополнительно редактировать, добиваясь выполнения остальных технических ограничений, не упоминаемых в этой работе.

Так же отметим, что программа работает достаточно быстро, что важно, потому что в реальной жизни требуется многократный запуск алгоритма. Видно, что скорость работы программы довольно сильно зависит от структуры конкретных начальных данных. Имеет значение соотношение между количеством подграфов в исходном графе и количеством входящих в них узлов. Например в случае, когда исходный граф состоит из малого числа подграфов, то время работы уменьшается за счет сокращения перебора дуг между подграфами(выборка 6).

(Тесты проводились на персональном компьютере).

5.3 Инструкция пользователю

Заключение

Автоматическое планирование металлургического производства является сложной задачей, которая до сих пор не реализована полностью ни на одном предприятии в мире. Сопутствующие ей математические постановки зачастую являются труднорешаемыми. За счет исследования технологических ограничений на конкретном предприятии удалось сформулировать математическую задачу планирования графика прокатки. Удалось разработать полиномиальный алгоритм её решения. Результат уже частично внедрён и продолжается внедрятся на ОАО "ММК". В реальной жизни существуют более общие и сложные постановки задачи планирования графиков прокатки, решение которых основывается на описанных в настоящей работе алгоритмах.

Литература

[1] *Леонова С.И.* Задача о формировании графика работы стана горячей прокатки, сбалансированного по составу продукции, 2016

[2] *Леонова С.И.* Разработка и исследование алгоритмов построениея экстремальных цепей в вершинно-взвешенных ориентированных графах. Частный алгоритм, 2014

[3] *Березин А.А.* Разработка и исследование алгоритмов построениея экстремальных цепей в вершинно-взвешенных ориентированных графах. Общий алгоритм, 2016

[4] *Dantzig G.B., Fulkerson D.R. , Johnson S.* On a linear programming combinatorial approach to the traveling salesman problem, Operations Research, 7 (1959), pp. 58–66

[5] *Balas E., Clarence H. M.*Combinatorial optimization in steel rolling. Workshop on Combinatorial Optimization in Science and Technology, April, 1991.

[6] *Lixin Tang, Jiyin Liu, Aiying Rong, Zihou Yang.* A review of planning and scheduling systems and methods for integrated steel production. European Journal of Operational Research. Volume 133, Issue 1, 16 August 2001, PP. 1–20.

[7] *Shixin Liu.* Model and Algorithm for Hot Rolling Batch Planning in Steel Plants. International Journal of Information and Management Sciences. 21 (2010), PP. 247-263.

Приложение

Пример исходного кода