О построении графиков прокатки

А.А. Березин, И.А. Вакула, С.И. Леонова

16 мая 2014 г.

Введение

В настоящей работе рассматривается вариант математической постановки задачи построения графиков прокатки партий на непрерывных станов горячей и холодной прокатки. В ряде работ (например, [2], [3]) приведены практические постановки задач совместного планирования непрерывной разливки стали и последующей горячей прокатки. Подробно обсуждаются различные варианты организации работы предприятия, когда разлитый металл полностью остывает, прежде, чем идет в нагревательные печи и на прокатку, либо остывает частично в виду наличия транспортного плеча, остывает мало и идет, как принято говорить на отечественных предприятиях, в прокатку горячим садом. Иллюстрируется эффект энергосбережния и прочее. Также в работах приводятся основные технологические ограничения, связанные с формированием графиков прокатки, которые используются в аналогичных условиях большинством предприятий. Приводятся некоторые математические модели, используемые при расчете графиков прокатки, для таких задач в иностранной литературе приняты названия "hot strip mill planning"и HRBPPs (hot rolling batch planning problems). Приводимые в работах модели представляют собой аналоги и обобщения задач коммивояжера TSP (Travelling Salesman Problem) и планирования маршрутов транспорта VRT (vehicle routing problem, [10]) стоимостью перемещения между узлами и доставкой ресурсов в узлы с ограничением по грузоподъемности машин. Модель PCTSP (Prize Collecting TSP), предложенная Баласом ([6], [7]). Задача PCVRP (Prize Collecting VRP, [9]). В них для каждой пары партий прокатки задана стоимость перехода с одной на другую, то есть стоимость того, что в графике прокатки они непосредственно следуют друг за другом, награда (приз) за то, что партия входит в построенный график прокатки, а также возможнен штраф за невхождение партии в график прокатки. В задаче PCVRP обходчиков может быть более одного, что фактически означает построение сразу нескольких графиков прокатки. Общим также является построение маршрутов с максимизацией получаемых призов, уменьшенных на сумму штрафов (весов) используемых дуг при переходе с одного узла (партии прокатки) на другой. Указанные задачи являются в общем случае трудноречаемыми.

Учитывая специфичный способ описания ограничений предшествования партий прокатки в графике прокатки, принятый на отечественных предприятиях (например, ОАО "ММК"), в настоящей работе задача построения графиков прокатки в своих базовых ограничениях сводится к задаче поиска простой цепи достаточно большого веса в вершинно-взвешенных ориентированных графах. Таким образом, из ценовой функции удаляется составляющая штрафов за переход, а сами переходы попадают в состав ограничений. Смежность узлов задается соотношением между ширинами и толщинами последовательно прокатываемых полос. Благодаря способу задания смежности удается построить алгоритм сложности ограниченной кубом числа узлов (партий) для построения простой цепи (графика прокатки) максимального веса. Хотя в общем случае задача поиска простой цепи максимального веса трудноречаема.

Поясним причину выбора критерия оптимизации. Оценки графиков прокатки связаны с особенностями организации производства на конкретном предприятии. Вместе с тем общим является то, что график прокатки должен содержать достаточно таких партий, чтобы суммарный вес/длина партий были достаточно велики. Более того, наличие "большого"технологически допустимого графика прокатки позволяет на следующем этапе в автоматическом или ручном режиме его отредактировать, чтобы получить желаемый результат по всей совокупности практических требований. Также в качестве веса партии может выступать величина, характеризующая не только ее физические характеристики, но и предпочтительность по срокам, видам продукции и т.п. По указанным причинам для получения оптимизационной постановки задачи естественно потребовать максимальности суммарного веса партий, входящих в монтаж (простую цепь в графе).

Постановка задачи

Обозначим через P множество партий $\{p_1, p_2, \dots, p_k\}$, где $k \in \mathbb{N}$. Пусть

- $w : P \to \mathbb{R}^+$ — ширина полос в партии (одинакова для всех полос);
- $t : P \to \mathbb{R}^+$ — толщина полос в партии (также одинакова);
- $l : P \to \mathbb{R}^+$ — суммарная длина полос в партии.
- $r : \mathbb{R}^+ \to \mathbb{R}^+$ — монотонно неубывающая функция, определяющая максимальную величину допустимой разности значений толщины полос между двумя соседними партиями;
- $\mathbb{W} \in \mathbb{R}^+$ — величина, определяющая максимальную величину допустимой разности значений ширины полос между двумя соседними партиями.

Две партии p и q из P могут непосредственно следовать друг за другом в графике прокатки в том и только в том случае, когда выполнено

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

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 длину:

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

Сформулируем задачу оптимизации — найти в $\mathbb{B}$ цепь максимальной длины:

Задача 1

$l(z) \to \max$

$z \in \mathbb{B}$

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.
- $W = \{w_p | p \in P\}$.
- $P_w = \{p | w_p = w\}$, где $w \in W$.
- G_w - граф, в котором $P(G_w) = P_w$.

Нетрудно видеть, что для решения задачи 1 достаточно искать наибольшую по длине цепь среди максимальных по включению цепей.

Метод решения

Для эффективного решения задачи авторами предложен следующий подход :

1. Разрабатывается алгоритм, на вход которого подается граф G' с тем же множеством узлов, что и у G, с меньшим множеством дуг и тем же множеством максимальных цепей, что и в исходном графе. Алгоритм возвращает цепь максимальной длины. Нетрудно видеть, что в частности можно взять $G' = G$.
2. Предлагается эффективная конструкция графа G', позволяющая существенно снизить время работы алгоритма.
3. Проводится анализ сложности алгоритма.
4. Приводятся результаты тестов и их анализ.

Алгоритм поиска цепи максимальной длины

Построение графа G'

Обозначим $\bar{G}$ - граф, полученный из исходного графа G удалением дуг pq, для которых существует путь $H = pe_1p' \dots p''e_kq$ такой, что $w_p > w_{p'}$ и $w_{p''} > w_q$.

Теорема 1. *Граф $\bar{G}$ содержит все максимальные по включению цепи графа G.*

Доказательство. Пусть H_m - некоторая максимальная по включению цепь в графе G. Предположим, что H_m не содержится в $\bar{G}$. То есть существует хотя бы одна дуга $pq \in H_m$, которая не содержится в $E(\bar{G})$. Значит, для этой дуги существует путь $H = pe_1p' \dots p''e_kq$, где $w_p > w_{p'}$ и $w_{p''} > w_q$. Это условие противоречит максимальности цепи H_m. Следовательно, любая максимальная цепь графа G содержится в графе $\bar{G}$. $\square$

Таким образом, в качестве G' можно взять граф $\bar{G}$.

Поиск максимальной цепи в одной ширине

Прежде чем формулировать общий алгоритм, приведем алгоритм построения максимальной по длине цепи в одной ширине.

Для произвольного упорядоченного набора узлов $H' = (p'_1, p'_2, \dots, p'_l)$ $\exists i : p'_i = \arg\max\limits_{p' \in H'} t'_p$. Тогда *крайней правой дугой* назовем дугу $p'_ip'_{i+1}$, если $t_{p'_{i+1}} \geq t_{p'_{i-1}}$, иначе дугу $p'_{i-1}p'_i$.

Аналогично *крайней левой дугой* H' назовем дугу $p'_ip'_{i+1}$, если $t_{p'_{i+1}} \leq t_{p'_{i-1}}$, иначе дугу $p'_{i-1}p'_i$, где $p'_i = \arg\min\limits_{p' \in H'} t'_p$.

Алгоритм 1 построения максимальной простой цепи в одной ширине.

Пусть заданы $w \in W$ и $s, t \in P_w$.

Алгоритм возвращает упорядоченный набор $H = (p_1, p_2, \dots, p_n)$, что $s = p_1 \to p_2 \to \dots \to p_n = t$ - максимальная по включению узлов (s, t) - цепь для заданной ширины w, или сообщение, что такую цепь построить нельзя.

1. Пусть

$H_0 = (p_1, p_2, \dots p_k)$, где $p_i \in P_w$ и $\forall i : t_s \leq t_{p_i} \leq t_{p_{i+1}} \leq t_t$.
$H_0^+ = \{p | t_p > t_t\}$.
$H_0^- = \{p | t_p < t_s\}$.

2. Проверить условие : $\forall i : p_i, p_{i+1} \in H_0 \Rightarrow p_i p_{i+1} \in E$.

   Если оно не выполнено, то нельзя построить путь максимальной длины.

3. Если $|H_0| > 2$, то

   (a) Если $H_i^+ \neq \emptyset$

   - $q_i = \arg\min\limits_{q \in H_i^+} t_q$, $p_j^i p_{j+1}^i$ - крайняя правая дуга $H_i = (p_1^i, p_2^i, \dots, p_{n_i}^i)$
   - Если $p_j^i q_i \in E$ и $q_i p_{j+1}^i \in E$, то
     $H_{i+1}^+ = H_i^+ \setminus \{q_i\}$,
     $H_{i+1} = (p_1^i, p_2^i, \dots, p_j^i, q_i, p_{j+1}^i, \dots, p_{n_i}^i)$,
     повторить 3a.
     Иначе $H_{i+1}^+ = \emptyset$.

   (b) Если $H_i^- \neq \emptyset$

   - $q_i = \arg\max\limits_{q \in H_i^-} t_q$, $p_j^i p_{j+1}^i$ - крайняя левая дуга $H_i = (p_1^i, p_2^i, \dots, p_{n_i}^i)$
   - Если $p_j^i q_i \in E$ и $q_i p_{j+1}^i \in E$, то
     $H_{i+1}^- = H_i^- \setminus \{q_i\}$,
     $H_{i+1} = (p_1^i, p_2^i, \dots, p_j^i, q_i, p_{j+1}^i, \dots, p_{n_i}^i)$,
     повторить 3b
     Иначе $H_{i+1}^- = \emptyset$.

   (c) $H = H_i$ - максимальная по включению цепь.

4. Если $|H_0| = 2$, то имеет значение, какой из списков H^+ или H^- первым добавлять в H. Поэтому сначала выполняются пункты 3a, 3b. Затем выполняются сначала 3b, а затем 3a. После чего получаются две максимальные по включению цепи. (Из них выбирается максимальная по длине).

5. В случае $s = t$ максимальная по длине простая цепь состоит из одного элемента s.

Теорема 2. *Алгоритм 1 строит максимальную по включению цепь.*

Доказательство. 1. Пусть H - цепь, построенная по алгоритму 1, а H_m - максимальная по включению цепь. Предположим, H - не максимальна, то есть существуют узлы из H_m, которые не попали в H.

2. Понятно, что если $\exists p \in H_m \setminus H$, то $t_p \notin [t_s, t_t]$, т. к. в противном случае p обязательно попадет в H.

3. Пусть $p'p''$ - крайняя правая дуга цепи H. И пусть для определенности $t_{p''} \geq t_{p'}$. Тогда в силу построения H и определения крайней правой дуги справедливо равенство $t_{p'} = \max\limits_{p \in H \setminus \{p''\}} t_p$.

4. Рассмотрим множества $V^+ = \{p | p \in H_m \setminus H, t_p > t_t\}$, $V^- = \{p | p \in H_m \setminus H, t_p < t_s\}$. Пусть V^+ не пусто и $q = \arg\min\limits_{p \in V^+} t_p$. Покажем, что q можно включить в H.

5. Заметим, что $H_m = V^- \cup H \cup V^+$. В силу того, что H_m - цепь из s в t, найдутся дуги из множества $V^- \cup H$ в V^+ и обратно. То есть найдутся узлы $v_1, v_2 \in V^- \cup H$ и $u_1, u_2 \in V^+$, что дуги v_1u_1 и u_2v_2 содержатся в H_m.

6. Из того, что $v_1u_1 \in H_m$ следует, что расстояние между t_{v_1} и t_{u_1} достаточно мало. Но расстояние между t_{v_1} и t_q еще меньше, поэтому дуга $v_1q \in E$. Аналогично, дуга $qv_2 \in E$.

7. В силу определения крайней правой дуги расстояние до t_q уменьшится, если заменить v_1 на p', а v_2 на p'', то есть $p'q \in E$ и $qp'' \in E$. Тогда q можно включить в H по алгоритму.

8. Таким образом, любой элемент из V^+, так же как и любой элемент из V^-, можно включить в H.

9. Следовательно, H - максимальная по включению узлов цепь. В силу пункта 4 алгоритма будет построена максимальная по длине цепь.

□

Поиск максимальной по длине цепи в графе G

Пусть $p \in P$. Для двух произвольных простых цепей $H_1 = p_1e_1p_2e_2 \ldots e_{n-1}p_n$ и $H_2 = q_1u_1q_2u_2 \ldots u_{m-1}q_m$, не имеющих общих узлов, *суммой* этих цепей назовем цепь $H = p_1 \dots p_n e q_1 \dots q_m$, если $p_n q_1 \in E$ или $p_n = q_1$, и обозначим $H = H_1 + H_2$.

Алгоритм 2 построения максимальной по длине простой цепи.

1. $w_0, w_1, \dots, w_z$ - всевозможные значения ширины, упорядоченные по убыванию.

2. На нулевом шаге

   - $P_{w_0} := \{p_1^0, p_2^0, \dots, p_{n_0}^0\}$.
   - $\forall r \in \overline{1, n_0}, \forall k \in \overline{1, n_0} : h_{rk}^0$ - максимальная по длине простая цепь, соединяющая p_r^0 и p_k^0, построенная алгоритмом 1.
   - $\forall m \in \overline{1, n_0} : H[p_m^0] := \arg \max\limits_{r \in \overline{1, n_0}} l(h_{rm}^0)$.
   - $H^0 := \arg \max\limits_{m \in \overline{1, n_0}} l(H[p_m^0])$.

3. На i-ом шаге

   - $P_{w_i} := \{p_1^i, p_2^i, \dots, p_{n_i}^i\}$.
   - $\forall r \in \overline{1, n_i}, \forall k \in \overline{1, n_i} : h_{rk}^i$ - максимальная по длине простая цепь из p_r^i в p_k^i, построенная алгоритмом 1.
   - $\forall r \in \overline{1, n_i}$: $H_r^i := \arg \max\limits_{p \in In^+(p_r^i)} l(H[p])$
   - $\forall m \in \overline{1, n_i}$: $H[p_m^i] := \arg \max\limits_{r \in \overline{1, n_i}} l(H_r^i + h_{rm}^i)$
   - $H^i := \arg\max\{l(H^{i-1}), l(H[p_1^i]), \dots, l(H[p_{n_i}^i])\}$.

4. $H = H^z$ - максимальный по длине путь.

Теорема 3. *Алгоритм 2 строит максимальную по длине цепь.*

Доказательство. Докажем, что для любого узла p построенная алгоритмом простая цепь $H[p]$ является максимальной по длине.

База индукции. Пусть p такой, что $w_p = w_0$. На нулевом шаге алгоритм строит максимальные по длине простые цепи в одной ширине в силу теоремы 2.

Шаг индукции. Пусть $\forall i \leq k$, $\forall p \in P_{w_i}$: $H[p]$ - максимальная по длине простая цепь, оканчивающаяся узлом p. Докажем, что $\forall q \in P_{w_{k+1}} : H[q]$ - максимальная по длине простая цепь с концом в q.

Пусть $q \in P_{w_{k+1}}$, $H[q]$ - цепь, построенная по алгоритму 2. Пусть также K - максимальная по длине простая цепь с концом в q. Предположим, что $l(K) > l(H[q])$.

$H[q] = p_1 \dots p_l e_l p_{l+1} \dots p_s e_s q$, где $w_{p_i} > w_{p_j} = w_q$ по всем $i \in \overline{1,l}, j \in \overline{l+1,s}$.

$K = q_1 \dots q_m e_m q_{m+1} \dots q_t e_t q$, где $w_{q_i} > w_{q_j} = w_q$ по всем $i \in \overline{1,m}, j \in \overline{m+1,t}$.

Тогда $l(H[q]) = l(p_1 \dots p_l) + l(p_{l+1} \dots p_s) + l(q)$,

и $l(K) = l(q_1 \dots q_m) + l(q_{m+1} \dots q_t) + l(q)$.

Рассмотрим цепь K. По предположению индукции $l(H[q_m]) \geq l(q_1 \dots q_m)$. Пусть h - максимальная по длине простая цепь, соединяющая q_{m+1} и q, построенная по алгоритму 1. Тогда $l(H[q]) \geq l(H[q_m]) + l(h) \geq$[по теореме 2]$\geq l(H[q_m]) + l(q_{m+1} \dots q_t) + l(q) \geq l(q_1 \dots q_m) + l(q_{m+1} \dots q_t) + l(q) = l(K)$. Таким образом $H[q]$ максимальная по длине простая цепь, заканчивающаяся в q.

□

Теорема 4. *Алгоритм 2 в худшем случае имеет сложность $O(k^3)$, где $k = |P|$.*

Доказательство. Пусть $n_i = |P_{w_i}|$, $m = |W|$. Алгоритм производит m итераций по количеству различных значений ширины. При построении максимального пути для двух фиксированных узлов, алгоритм в худшем случае просматривает один раз отсортированный по толщине список узлов в данной ширине. То есть, на каждом шаге список узлов сортируется за $n_i \log n_i$, после чего перебираются попарно узлы (n_i^2) и для них строится максимальный путь (n), и для каждого узла перебирается список его предков.

Тогда сложность алгоритма 2 не превосходит $(\sum_{i=1}^{m}(\sum_{p \in P_{w_i}} |In^+(p)| + n_i \log n_i + n_i^3))$, где C — некоторая константа, что в худшем случае, когда все узлы находятся в одной ширине есть $O(k^3)$. □

Список литературы

[1] Постановки задач планирования прокатки:

[2] 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.

[3] P. Cowling, W. Rezig. Integration of continuous caster and hot strip mill planning for steel production. Journal of Scheduling, Volume 3, Issue 4, July/August 2000, PP. 185–208.

[4] Конкретные модели и алгоритмы:

[5] PCTSP

[6] E. Balas, The prize collecting travelling salesman problem, Networks 19 (1989) 621-636.

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

[8] PCVRT

[9] 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.

[10] G. B. Dantzig and J. H. Ramser. Management Science, Vol. 6, No. 1 (Oct., 1959), pp. 80-91.

[11] Andreas Stenger, Michael Schneider, Dominik Goeke. The prize-collecting vehicle routing problem with single and multiple depots and non-linear cost. EURO Journal on Transportation and Logistics, May 2013, Volume 2, Issue 1-2, pp 57-87.

[12] Исследование задачи построения длиннейшего пути в графе

[13] K. Ioannidou, S. D. Nikolopoulos. The Longest Path Problem is Polynomial on Cocomparability Graphs. Algorithmica. January 2013, Volume 65, Issue 1, pp 177-205.