ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель—июнь 2007. Серия 1. Том 14, № 2. 25–46

УДК 519.8

УЛУЧШЕННАЯ АППРОКСИМАЦИОННАЯ СХЕМА ДЛЯ ЗАДАЧИ ДЖОНСОНА С ПАРАЛЛЕЛЬНЫМИ МАШИНАМИ*)

С. В. Севастьянов

Рассматривается обобщение NP-трудной задачи Джонсона на случай идентичных параллельных машин на каждой стадии выполнения работ. В условиях, когда число стадий ограничено константой, а общее число машин является частью входа, предложена новая полиномиальная аппроксимационная схема, имеющая лучшую оценку сложности по сравнению с ранее известной.

Введение

Рассматривается задача построения кратчайшего расписания выполнения заданного множества работ в многостадийной производственной системе, описываемой следующим образом.

Имеется множество работ $\{J_1, \dots, J_n\} \doteq \mathcal{J}_n$ и s специализированных цехов $\mathcal{M}_1, \dots, \mathcal{M}_s$. Цех $\mathcal{M}_q$ состоит из m_q идентичных машин; $m = \sum\limits_{q=1}^{s} m_q$ — общее число машин; машины нумеруются числами от 1 до m: $M_1, \dots, M_m$. Каждая работа J_k состоит из s операций $O_{1k}, \dots, O_{sk}$, выполняемых в порядке $O_{1k} \to \dots \to O_{sk}$, где $O' \to O''$ означает, что операция O'' может начаться не раньше, чем закончится операция O'. Операция O_{qk} может быть выполнена на любой из машин цеха $\mathcal{M}_q$ за время p_{qk}. При этом процесс выполнения каждой операции должен быть непрерывным: начав выполняться в момент $s(O_{qk})$, она завершается в момент $c(O_{qk}) = s(O_{qk}) + p_{qk}$, полностью занимая в интервале $(s(O_{qk}), s(O_{qk}) + p_{qk})$ назначенную ей машину $M_{\mu(O_{qk})} \in \mathcal{M}_q$. (В этом интервале машина $M_{\mu(O_{qk})}$ не может выполнять другие операции, даже если последние имеют нулевую длительность.)

Ясно, что всякое допустимое расписание выполнения работ в такой системе однозначно задаётся набором чисел $S \doteq \{s(O_{qk}), \mu(O_{qk}) \mid q =$

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 05–01–00960).

$1,\dots,s;\ k=1,\dots,n\}$, который далее будем именовать *расписанием.* Длиной расписания S будем называть величину $C_{\max}(S) \doteq \max\limits_{q,k}\ c(O_{qk})$, т. е. момент завершения наиболее поздней операции. Для описанной выше системы требуется найти допустимое расписание наименьшей длины.

Данная задача, обозначаемая согласно общепринятой классификации задач [6] как $\mathbf{F(P)}||\boldsymbol{C}_{\mathbf{max}}$, является обобщением двух классических задач: *задачи Джонсона* $\mathbf{F}||\boldsymbol{C}_{\mathbf{max}}$ (в которой каждый цех $\mathcal{M}_q$ представлен единственной машиной M_q) и задачи $\mathbf{P}||\boldsymbol{C}_{\mathbf{max}}$ о выполнении работ на параллельных идентичных машинах. Для обеих задач установлена NP-трудность в сильном смысле: для задачи Джонсона уже в случае трёх машин [8], а для $\mathbf{P}||\boldsymbol{C}_{\mathbf{max}}$ в случае растущего числа машин [7] (являющегося переменной, или, как ещё говорят, *частью входа*). В то же время при любом фиксированном числе машин m $(m>1)$ задача $\mathbf{Pm}||\boldsymbol{C}_{\mathbf{max}}$ является NP-трудной лишь в простом смысле и допускает псевдополиномиальное решение.

Не ставя целью привести здесь полный обзор литературы, посвящённой анализу этих двух задач, упомянем лишь результаты, связанные с построением для них полиномиальных аппроксимационных схем.

Наилучшим в этом смысле результатом, на который можно надеяться для общей задачи $\mathbf{P}||\boldsymbol{C}_{\mathbf{max}}$ (с переменным числом машин), является полиномиальная аппроксимационная схема (PTAS), построенная в [11]. Вполне полиномиальной схемы (FPTAS) для этой задачи не существует (при условии верности гипотезы P$\neq$ NP), что следует из NP-трудности данной задачи в сильном смысле. В то же время при фиксированном числе машин m схема FPTAS для задачи $\mathbf{Pm}||\boldsymbol{C}_{\mathbf{max}}$ построена в [18].

Первая PTAS для задачи $\mathbf{Fm}||\boldsymbol{C}_{\mathbf{max}}$ при фиксированном числе машин m построена в [10]. Временна́я сложность схемы Холл имеет порядок временно́й сложности решения линейной программы размера n. Позднее в [14] была построена существенно более эффективная схема для более общей задачи job shop с константным числом машин и константным ограничением на число операций каждой работы. В то же время в [22] показана невозможность построения PTAS для задачи $\mathbf{F}||\boldsymbol{C}_{\mathbf{max}}$ с переменным числом машин (при условии справедливости гипотезы P$\neq$ NP).

Ясно, что совмещение двух NP-трудных задач не может привести к более лёгкой задаче. Поэтому уже двухстадийная задача $\mathbf{F2(P)}||\boldsymbol{C}_{\mathbf{max}}$ NP-трудна в сильном смысле. Более того, она остаётся таковой даже в случае, когда в одном из двух цехов имеется лишь одна машина, а в другом — две [12]. Таким образом, уже для двухстадийной задачи разработка эффективных приближённых алгоритмов является актуальным направлением. Построению приближённых алгоритмов для двухстадийных систем flow shop посвящены работы [3, 4, 5, 9, 16, 17, 21], в которых помимо эмпирических методов проверки эффективности алгоритмов выводились также априорные оценки их относительной погрешности в худшем случае.

Одной из наиболее ранних статей, в которых рассматриваются k-стадийные системы flow shop с параллельными машинами при $k > 2$, по-видимому, является статья [1]. В ней на основе метода компактного суммирования векторов был разработан асимптотически точный алгоритм приближённого решения с абсолютной оценкой точности, гарантированной в худшем случае (однако лишь в частном случае, когда все цеха имеют одинаковое число машин $m_q \equiv m'$). Это направление получило дальнейшее развитие в работах [2, 15, 20], в которых аналогичные асимптотически точные алгоритмы были разработаны как для задачи $\mathbf{F(P)}||\boldsymbol{C}_{\mathbf{max}}$ в общем случае, так и для более общей задачи $\mathbf{F(Q)}||\boldsymbol{C}_{\mathbf{max}}$ с неидентичными (*неподобными*) машинами в каждом цехе.

Перейдём, наконец, к вопросу о существовании аппроксимационных схем для рассматриваемой задачи $\mathbf{F(P)}||\boldsymbol{C}_{\mathbf{max}}$. Уже упоминавшаяся выше схема Холл, разработанная для задачи $\mathbf{Fm}||\boldsymbol{C}_{\mathbf{max}}$, как утверждается автором, допускает несложное обобщение и на случай идентичных параллельных машин в каждом цехе при условии, что число цехов и число машин в каждом цехе ограничены константой (данная задача обозначается через $\mathbf{Fs(Pm)}||\boldsymbol{C}_{\mathbf{max}}$). В этой же работе автором был поставлен вопрос о существовании PTAS для данной задачи в случае, когда число цехов ограничено константой, но число машин в каждом цехе является переменной величиной. Следует заметить, что такая постановка вопроса является правомерной, поскольку из [22] следует «неперспективность» попыток построения PTAS для данной задачи в случае переменного числа цехов (даже если в каждом цехе имеется всего одна машина, а каждая работа имеет не более трёх операций).

Первый шаг в попытке ответа на поставленный вопрос был сделан в работе [19], в которой была построена PTAS для двухстадийной задачи $\mathbf{F2(P)}||\boldsymbol{C}_{\mathbf{max}}$. Вскоре после этого К. Янсеном и М. Свириденко был дан принципиальный ответ на вопрос о существовании PTAS и для задачи $\mathbf{Fs(P)}||\boldsymbol{C}_{\mathbf{max}}$: в [13] они представили первую аппроксимационную схему для данной задачи, имеющую полиномиальную временну́ю сложность при произвольном, но *фиксированном* числе цехов и переменном числе машин.

Более точно, временна́я сложность $(1+\varepsilon)$-приближающего алгоритма оценена в [13] величиной, содержащей «четырёхэтажный» сомножитель $n^{10^{65\,s^6/\varepsilon^3}}$, где n — число работ. В этой оценке показатель степени «полинома» от n выражается экспонентой от фиксированного параметра задачи (числа цехов s) и от $1/\varepsilon$. Однако, хотя в данной работе даётся принципиальный ответ на вопрос о существовании PTAS, остаётся открытым вопрос о возможности построения менее трудоёмких аппроксимационных схем. В идеале хотелось бы получить минимальную (по порядку сложности) аппроксимационную схему, теоретически возможную для исследуемой задачи. В настоящей статье в качестве дальнейшего шага в анализе вычислительной сложности данной задачи построена полиномиальная от n PTAS, когда показатель степени при n является уже не экспонентой, а полиномом от s и $1/\varepsilon$.

1. Исходные понятия и обозначения

«Быстрота» аппроксимационных схем обычно достигается за счёт того, что такой объект как расписание заменяется более простым объектом — *конфигурацией расписания*. Последнюю можно рассматривать как «грубый набросок» расписания, «приближённое расписание». Конфигурации допускают гораздо меньшее разнообразие, что позволяет перебрать их за приемлемое время. (В то время как перебор всевозможных расписаний является безнадежным делом для NP-трудных задач.) Однако заменяя расписания конфигурациями, мы теряем в точности алгоритма. В результате «сделки» между точностью приближения расписаний их конфигурациями и оценкой временно́й сложности алгоритма и получается полиномиальная по времени аппроксимационная схема решения задачи.

Коль скоро «секрет скорости» любой аппроксимационной схемы кроется в используемых ею конфигурациях, то в них же скрываются и главные резервы дальнейшего совершенствования схем. Полученное нами существенное улучшение временно́й сложности PTAS также достигнуто за счёт построения более экономной конфигурации расписания, описание которой приводится в разделе 4.

Прежде чем перейти к определению конфигурации расписания, выполним несколько последовательных упрощающих трансформаций расписания по схеме: допустимое расписание → допустимое округлённое расписание (ДОР) → специальное ослабленное расписание (СОР) → конфигурация расписания.

Пусть $\hat{L}_q = \sum\limits_{j=1}^{n} p_{qj}/m_q$ — средняя загрузка цеха $\mathcal{M}_q$, $p_q^{\max} = \max\limits_{j=1,\dots,n} p_{qj}$

— максимальная длительность операции в цехе $\mathcal{M}_q$. Нетрудно видеть, что оптимум $C^*_{\max}$ удовлетворяет оценкам

$$\underline{C} \doteq \max_{q=1,\ldots,s} \max\{\hat{L}_q, p_q^{\max}\} \leqslant C^*_{\max} \leqslant \sum_{q=1}^{s} (\hat{L}_q + p_q^{\max}) \doteq \overline{C},$$

где верхняя оценка оптимума $\overline{C}$ отличается от нижней оценки $\underline{C}$ не более чем в $2s$ раз. Уменьшив длительности всех операций в $2s\underline{C}$ раз, получим эквивалентную задачу, оптимум которой удовлетворяет оценкам

$$1/(2s) \leqslant C^*_{\max} \leqslant 1. \tag{1}$$

Далее будем считать, что рассматриваемая задача удовлетворяет (1).

2. Допустимое округлённое расписание

Пусть требуемая точность решения задачи задана числом $\varepsilon = 1/t$, где t — целое. В конструкции $(1+\varepsilon)$-приближающего алгоритма будут использоваться два неотрицательных параметра κ и δ такие, что $\kappa = 1/\eta$, $\delta = 1/\sqrt{\nu}$, где η и ν — целые. В результате оптимизации этих параметров для них будут выбраны значения

$$\delta = \frac{\varepsilon}{2s+5};\ \kappa = \frac{\delta}{2s}. \tag{2}$$

На числовой полуоси $\mathbb{R}^+$ определим три дискретные решётки чисел: *сегментную сетку* с шагом κ, *крупную сетку* с шагом $\kappa' \doteq \kappa\delta^2 = \kappa/\nu$ и *мелкую сетку* с шагом $\delta^* \doteq \kappa\delta/(ns)$. Элементы решёток будем называть *узлами.* Все три решётки содержат число нуль. При этом сегментная сетка содержится в крупной, поскольку κ нацело делится на κ'. Интервалы, на которые сегментная и крупная решётки делят числовую полуось, будем называть *сегментами* и *κ'-интервалами* соответственно. Все сегменты и κ'-интервалы считаем занумерованными слева направо числами $\{1, 2, \ldots\}$ Каждый сегмент содержит ν κ'-интервалов.

Все операции разделим на два подмножества: *большие* (с длительностью $p_{qj} > \kappa\delta$) и *малые* операции (с длительностью $p_{qj} \leqslant \kappa\delta$).

Пусть задан вход задачи I, который характеризуется набором длительностей $\{p_{qj} \mid q = 1, \ldots, s;\ j = 1, \ldots, n\}$. Вход I' с набором длительностей $\{p'_{qj} \mid q = 1, \ldots, s;\ j = 1, \ldots, n\}$ будем называть *округлением* входа I, если

а) для каждой большой операции O_{qj} длительность p'_{qj} получается округлением p_{qj} вверх до ближайшего узла крупной сетки;

б) для каждой малой операции O_{qj} длительность p'_{qj} получается округлением p_{qj} вверх до ближайшего узла мелкой сетки. Поскольку число $\kappa\delta$ (т. е. граница, отделяющая малые и большие операции) является узлом мелкой сетки, то никакая малая операция после округления не станет большой (как, впрочем, и наоборот). Таким образом, *множества малых операций* входов I и I' *совпадают*.

Ясно, что всякое допустимое расписание округлённого входа I' является также допустимым для исходного входа I.

Допустимое расписание округления I' будем называть *допустимым округлённым расписанием* (ДОР) входа I, если каждая большая операция начинается в узле крупной сетки (тем самым занимая целое число κ'-интервалов).

Лемма 1. *Существует* ДОР S' *такое, что*

$$C_{\max}(S') \leqslant (1+2\delta)C^*_{\max} + \kappa\delta. \tag{3}$$

Доказательство. Пусть S_{opt} — оптимальное расписание для входа I. В этом расписании, в силу ограничения несовместности любых двух операций, выполняемых на одной машине, операции на каждой машине естественным образом выстраиваются в цепочку. Для выполняемых одновременно операций нулевой длины этот порядок доопределяем произвольным способом, считая таким образом, что на множестве операций каждой машины определён строгий порядок. После этого определяем орграф $G = (\mathcal{O}, U)$, в котором множество вершин совпадает с множеством операций, а дуга (O', O'') принадлежит U в двух случаях: 1) если обе операции выполняются на одной машине и в расписании S_{opt} операция O' предшествует операции O''; 2) если обе операции принадлежат одной работе и O' непосредственно предшествует операции O''.

Определим задачу сетевого планирования $\mathcal{P}(G)$ с множеством операций $\mathcal{O}$, графом предшествований G и критерием *минимум длины расписания*. Так как расписание S_{opt} является допустимым для этой задачи, а всякое допустимое расписание задачи $\mathcal{P}(G)$ является допустимым и для исходной задачи, то любое оптимальное расписание задачи $\mathcal{P}(G)$ является оптимальным и для исходной задачи.

Как известно из сетевого планирования, среди оптимальных расписаний задачи $\mathcal{P}(G)$ существует так называемое *раннее* (или *активное*) расписание S, в котором момент начала каждой операции O' является минимально возможным и совпадает с длиной наиболее длинного пути в графе G, который ведёт в вершину O'. В рассматриваемом случае (когда длины дуг графа G равны нулю) это обеспечивает следующее свойство (будем называть его свойством *активности*): *момент начала каждой операции O' равен сумме длин операций наиболее длинного пути в графе G, ведущего в вершину O'*. Без нарушения общности можем считать, что рассматриваемое оптимальное расписание S_{opt} обладает свойством активности.

Сначала построим раннее расписание $\tilde{S}$ для округлённого примера I', наследуя из расписания S_{opt} распределение операций по машинам и порядок выполнения операций, задаваемый графом G. Оценим увеличение длины расписания $\Delta_1 = C_{\max}(\tilde{S}) - C^*_{\max}$ в результате округления входа I. Так как длина расписания $\tilde{S}$ совпадает с длиной критического пути в графе G, достаточно оценить увеличение длины произвольного пути P в графе G.

Пусть в пути P содержится k_ℓ больших и k_s малых операций. Так как каждая большая операция O_{ij} имеет длину $p_{ij} > \kappa\delta$, а общее число операций равно ns, то

$$k_\ell < C^*_{\max}/\kappa\delta; \quad k_s \leqslant ns. \tag{4}$$

Отсюда округление операций даёт увеличение длины пути P не более чем на величину $k_\ell\kappa' + k_s\delta^* \leqslant C^*_{\max}\delta + \kappa\delta$, и значит, $\Delta_1 \leqslant C^*_{\max}\delta + \kappa\delta$.

Наконец, строим ДОР S', для чего запускаем алгоритм построения раннего расписания по заданному графу G. В этом алгоритме порядок вычисления величин $s(O_{qk})$ согласован с порядком вершин в графе G следующим образом (назовём это *свойством согласованности*): если $O' \prec O''$ в графе G, то $s(O')$ вычисляется раньше, чем $s(O'')$. В работу алгоритма внесём следующую поправку: при вычислении величины $s(O_{qk})$ для какой-либо большой операции O_{qk} будем округлять её вверх до ближайшего узла крупной сетки. Тем самым, как бы добавляется новая операция длины $< \kappa'$ в сеть G непосредственно перед каждой большой операцией.

Ясно, что каждое добавление новой операции (непосредственно перед какой-то большой операцией O_{qk}) может повлиять не только на момент начала самой операции O_{qk}, но и на моменты начал других операций, следующих за O_{qk} в графе G. Однако благодаря свойству согласованности каждое добавление новой операции не вызывает небходимости пересчёта уже найденных к данному моменту величин $s(O_{qk})$.

Поскольку с учётом добавления новых операций длина пути P может увеличиться не более чем на величину $k_\ell\kappa' \leqslant C^*_{\max}\delta$, это приводит к увеличению длины расписания не более чем на $\Delta_2 = C_{\max}(S') - C_{\max}(\tilde{S}) \leqslant C^*_{\max}\delta$.

Суммируя приведённые выше оценки, получаем, что $C_{\max}(S')-C^*_{\max} \leqslant \Delta_1 + \Delta_2 \leqslant \kappa\delta + C^*_{\max} \cdot 2\delta$, что доказывает (2). Лемма 1 доказана.

Из леммы 1 с учётом (1) и (2) получаем, что оптимальное ДОР лежит в первых η' сегментах, где $\eta' = \lceil \frac{1+2\delta}{\kappa} + \delta \rceil = \eta + 4s + 1$.

Заметим также, что согласно определению ДОР каждая большая операция O_{ij} целиком занимает $\tilde{p}_{ij} \doteq p'_{ij}/\kappa'$ κ'-интервалов: от интервала с номером $\tilde{s}(O_{ij})$ до интервала $\tilde{s}(O_{ij}) + \tilde{p}_{ij} - 1$.

3. Специальное ослабленное расписание

Пусть задано некоторое число $k \in \{1, \dots, n\}$. *Специальное ослабленное расписание* (СОР) $\tilde{S}$ для работ $J_1, \dots, J_k$, во-первых, является округлённым. Во-вторых, оно обладает «специальным свойством»: в каждом сегменте I_g большие операции занимают некоторый начальный и некоторый конечный отрезки сегмента на каждой машине (эти отрезки могут быть пустыми, либо, наоборот, объединение этих отрезков может покрывать весь сегмент). В-третьих, расписание $\tilde{S}$ удовлетворяет «ослабленным» ограничениям предшествования: для любых двух последовательных операций $O_{q-1,j}$ и O_{qj} какой-либо работы J_j вместо ограничения $c_{q-1,j} - s_{qj} \leqslant 0$ выполняется более слабое ограничение $c_{q-1,j} - s_{qj} \leqslant \Delta$ для достаточно малого Δ, значение которого будет уточнено позже.

В строгом смысле СОР $\tilde{S}$ является расписанием лишь для больших операций: для каждой большой операции O_{qj} задана запись $\tilde{S}(O_{qj})$ в виде пары $(\tilde{s}(O_{qj}), \tilde{\mu}(O_{qj}))$, где $\tilde{s}(O_{qj})$ — номер κ'-интервала, в котором O_{qj} начинает выполняться в расписании $\tilde{S}$, $\tilde{\mu}(O_{qj})$ — номер машины, на которой выполняется O_{qj}. Для каждой малой операции O_{qj} запись $\tilde{S}(O_{qj})$ состоит из единственного числа $\hat{s}(O_{qj})$ — номера сегмента, к которому «приписана» операция O_{qj}.

По величине $\tilde{s}(O_{qj})$ и длительности p_{qj} однозначно определяются номера $\hat{s}(O_{qj})$ и $\hat{c}(O_{qj})$ сегментов, в которых начинает и заканчивает выполняться большая операция O_{qj} в расписании $\tilde{S}$. Для малых операций O_{qj} полагаем $\hat{c}(O_{qj}) = \hat{s}(O_{qj})$. Тогда семейство записей $\tilde{S} = \{\tilde{S}(O_{qj}) \mid q = 1, \dots, s;\ j = 1, \dots, k\}$ является *специальным ослабленным расписанием* (СОР) для работ $J_1, \dots, J_k$, если выполнены следующие требования:

($\tilde{\text{a}}$). Для каждой машины интервалы выполняемых на ней больших операций не пересекаются.

($\tilde{\text{b}}$). Для каждой машины и каждого сегмента I_g совокупность моментов времени, в которых машина не занята большими операциями, представляет собой связный отрезок, составленный из целого числа κ'-интервалов.

($\tilde{\text{c}}$). Для любой работы J_j и двух её последовательных операций $O_{q-1,j}$, O_{qj} выполняется неравенство $\hat{c}(O_{q-1,j}) \leqslant \hat{s}(O_{qj})$.

($\tilde{\text{d}}$). Для каждого цеха $\mathcal{M}_q$, $q = 1, \ldots, s$, и каждого сегмента I_g, $g = 1, \ldots, \eta'$, выполняется соотношение

$$l_{qg} + \kappa' \cdot \sum_{M_i \in \mathcal{M}_q} P^i_{qg} \leqslant \kappa m_q + \kappa\delta m'_{qg}, \tag{5}$$

где $l_{qg} = \sum p'_{qj}$ — сумма округлённых длительностей малых операций O_{qj}, приписанных к сегменту I_g (т. е. таких, что $\hat{s}(O_{qj}) = g$); P^i_{qg} — суммарное число κ'-интервалов, занятых большими операциями на машине $M_i \in \mathcal{M}_q$ в сегменте I_g; m_q — общее число машин цеха $\mathcal{M}_q$; m'_{qg} — число машин цеха $\mathcal{M}_q$, не полностью занятых большими операциями в сегменте I_g (т. е. число машин M_i таких, что $P^i_{qg} < \nu$).

Длиной СОР $\tilde{S}$ называем величину $\hat{C}(\tilde{S}) = \kappa \cdot \max\limits_{\{O_{qj}\}} \hat{c}(O_{qj})$. Через $\tilde{S}^*$ обозначим СОР наименьшей длины.

Лемма 2. *Длина кратчайшего* СОР $\tilde{S}^*$ *работ* $J_1, \ldots, J_n$ *удовлетворяет оценкам*:

$$\hat{C}(\tilde{S}^*) < C^*_{\max}(1 + 2\delta) + \kappa\delta + \kappa, \tag{6}$$

$$\hat{C}(\tilde{S}^*) \leqslant \eta'\kappa. \tag{7}$$

Доказательство. Возьмём оптимальное ДОР S', длина которого оценивается в лемме 1, и по нему построим СОР $\tilde{S}$ следующим образом:

а) для каждой машины M_i и сегмента I_g все большие операции, выполняющиеся на M_i целиком внутри сегмента I_g, сдвинем до упора влево: каждая большая операция упрётся либо в предыдущую большую операцию, либо в начало сегмента I_g.

б) каждую малую операцию O_{qj} припишем к тому сегменту $\hat{s}(O_{qj})$, в котором она начинается в расписании S'; положим $\hat{c}(O_{qj}) := \hat{s}(O_{qj})$.

Для полученного СОР $\tilde{S}$ требования ($\tilde{\text{a}}$) и ($\tilde{\text{b}}$) выполняются по построению. Требование ($\tilde{\text{c}}$) выполняется, так как оно выполнялось в расписании S', величины $\hat{s}(O_{qj})$ и $\hat{c}(O_{qj})$ для больших операций и $\hat{s}(O_{qj})$ для малых операций остались теми же, а величины $\hat{c}(O_{qj})$ для малых операций не увеличились. Наконец, для доказательства выполнимости требования ($\tilde{\text{d}}$) отметим, что соотношение (4) выполнялось в расписании S'. (Это соотношение отражает тот факт, что «приписанная» цеху q в сегменте g нагрузка может немного превосходить «производительность» цеха за счёт «переходных» малых операций, начинающихся в сегменте

g и заканчивающихся в сегменте $g+1$. Хотя в расписании S' некоторая доля нагрузки этих операций на машину приходится на следующий сегмент, приписана она будет целиком сегменту g.) Поскольку при переходе от S' к $\tilde{S}$ величины P^i_{qg} и m'_{qg} остаются неизменными, сохраняется и выполнимость требования $(\tilde{\mathrm{d}})$.

Так как $\hat{C}(\tilde{S})$ совпадает с моментом окончания последнего сегмента, в котором начинает выполняться какая-либо малая операция либо заканчивает выполняться большая операция в расписании S', то справедлива оценка $\hat{C}(\tilde{S}) < C_{\max}(S') + \kappa$, из которой с учётом (2) следует оценка (5). Оценка (6) следует из того, что СОР $\tilde{S}$, построенное по оптимальному ДОР, располагается в тех же η' сегментах. Таким образом, кратчайшее СОР существует и удовлетворяет оценкам (5) и (6). Лемма 2 доказана.

Так как в алгоритме $\mathcal{A}_\varepsilon$ (который будет описан позже) предполагается нахождение кратчайшего СОР $\tilde{S}^*$ (с последующим преобразованием его в $(1+\varepsilon)$-приближённое расписание), то далее будем рассматривать лишь такие СОР, которые помещаются в η' сегментах.

4. Конфигурация расписания

Пусть имеется СОР $\tilde{S}$ для множества работ $\{J_1,\ldots,J_k\}$, покрываемое η' сегментами. Определим конфигурацию $\Theta(\tilde{S})$ этого расписания следующим образом.

Для начала определим конфигурацию машины M_i, $i=1,\ldots,m$, в сегменте I_g, $g=1,\ldots,\eta'$, как тройку неотрицательных чисел (α,β,γ), обозначающих число κ'-интервалов, занятых на машине M_i большими операциями в левой, средней и правой частях сегмента I_g соответственно. При этом левая часть сегмента, состоящая из α κ'-интервалов, покрывается единственной операцией, начинающейся в сегменте $g'<g$ и заканчивающейся в сегменте $g''\geqslant g$; правая часть сегмента, состоящая из γ κ'-интервалов, покрывается единственной операцией, начинающейся в сегменте g и заканчивающейся в сегменте $g''>g$; наконец, среднюю часть составляют большие операции, целиком выполняющиеся в сегменте I_g. Величины α,β и γ принимают значения из множества $\{0,1,\ldots,\nu\}$ и удовлетворяют неравенству $\alpha+\beta+\gamma\leqslant\nu$.

Тройку $\Omega=(\alpha,\beta,\gamma)$, удовлетворяющую указанным требованиям, будем называть *конфигурацией сегмент-машина*, или СМ-*конфигурацией*.

Пусть T — число всевозможных СМ-конфигураций. Оно равно числу различных упорядоченных разбиений натурального числа ν на 4 натуральных слагаемых и выражается числом сочетаний из $\nu+3$ по 3, т. е.

$$T=\frac{(\nu+3)!}{\nu!\cdot 3!}=\frac{1}{6}(\nu+1)(\nu+2)(\nu+3)=\frac{\nu^3-\nu}{6}+(\nu+1)^2.$$

В дальнейшем будем использовать оценку, справедливую для δ, определенной согласно (2) (для такого δ при любых $\varepsilon \leqslant 1$ и $s \geqslant 1$ выполнено неравенство $\nu \geqslant 49$):

$$T + 2 \leqslant \frac{\nu^3}{5} = \frac{1}{5\delta^6}. \tag{8}$$

Все СМ-конфигурации $\Omega_i = (\alpha_i, \beta_i, \gamma_i)$ перенумеруем числами $i = 1, \ldots, T$ в порядке лексикографического возрастания векторов $(d_i, \alpha_i, \beta_i, \gamma_i)$, где $d_i = \alpha_i + \beta_i + \gamma_i$. Таким образом, $\Omega_1 = (0, 0, 0)$, $\Omega_T = (\nu, 0, 0)$. Пусть $T' = \max\{i \,|\, d_i < \nu\}$ — номер последней конфигурации, при которой машина в данном сегменте не полностью занята большими операциями (имеет «неполную» конфигурацию).

Конфигурацией $\theta^{qg}(\tilde{S})$ *цеха* $\mathcal{M}_q$ в сегменте I_g назовём вектор целых неотрицательных чисел $\theta^{qg}(\tilde{S}) = (m_1^{qg}, \ldots, m_T^{qg}; L^{qg})$, где m_i^{qg} — число машин цеха $\mathcal{M}_q$, имеющих СМ-конфигурацию Ω_i в сегменте I_g расписания $\tilde{S}$; $L^{qg} = l_{qg}/\delta^*$ — целое число, обозначающее занятость цеха $\mathcal{M}_q$ в сегменте I_g малыми операциями. Очевидно, что

$$\sum_{i=1}^{T} m_i^{qg} = m_q \text{ при любых } q = 1, \ldots, s;\ g = 1, \ldots, \eta'. \tag{9}$$

Из (4) также следует, что при любых $q = 1, \ldots, s$; $g = 1, \ldots, \eta'$ справедливо неравенство

$$\delta^* L^{qg} + \kappa' \cdot \sum_{i=1}^{T} m_i^{qg} d_i \leqslant \kappa m_q + \kappa\delta m'_{qg}, \tag{10}$$

где $d_i = \alpha_i + \beta_i + \gamma_i,\ m'_{qg} = \sum\limits_{i=1}^{T'} m_i^{qg}$.

Наконец, *конфигурацией* $\Theta(\tilde{S})$ расписания $\tilde{S}$ назовём семейство $\Theta(\tilde{S}) = \{\theta^{qg}(\tilde{S}) \,|\, q = 1, \ldots, s;\ g = 1, \ldots, \eta'\}$ конфигураций цехов в сегментах. Семейство $\Theta = \{\theta^{qg} \,|\, q = 1, \ldots, s;\ g = 1, \ldots, \eta'\}$ векторов $\theta^{qg} = (m_1^{qg}, \ldots, m_T^{qg}; L^{qg})$, удовлетворяющих (8) и (9), будем называть *абстрактной конфигурацией*. Множество всех абстрактных конфигураций обозначим через $\boldsymbol{\Theta}$. Очевидно, конфигурация $\Theta(\tilde{S})$ любого СОР $\tilde{S}$ для множества работ $\{J_1, \ldots, J_k\}$ является абстрактной конфигурацией. Обратное, вообще говоря, неверно, т. е. не всякая абстрактная конфигурация $\Theta \in \boldsymbol{\Theta}$ является конфигурацией какого-либо СОР для множества работ $\{J_1, \ldots, J_k\}$.

Конфигурацию $\Theta' \in \boldsymbol{\Theta}$ будем называть *k-реализуемой* (или просто *реализуемой*), если она является конфигурацией какого-либо СОР

$\tilde{S}$ для множества работ $\{J_1, \ldots, J_k\}$: $\Theta' = \Theta(\tilde{S})$. Определим также *пустую* конфигурацию цеха $\mathcal{M}_q$ в сегменте I_g как $(T+1)$-мерный вектор $\theta^q_\varnothing \doteq (m_q, 0, \ldots, 0; 0)$, и *пустую абстрактную конфигурацию* $\Theta_\varnothing = \{\theta^{qg}\}$ такую, что $\theta^{qg} = \theta^q_\varnothing$, $q = 1, \ldots, s$; $g = 1, \ldots, \eta'$. *Длиной* $C(\Theta)$ непустой конфигурации $\Theta = \{\theta^{qg}\}$ назовём величину

$$C(\Theta) \doteq \kappa \cdot \max\{g \mid \text{существует } q \in \{1, \ldots, s\} \text{ такое, что } \theta^{qg} \neq \theta^q_\varnothing\} \doteq \kappa \cdot g^*(\Theta).$$

Очевидно, если $\Theta' = \Theta(\tilde{S})$, то $C(\Theta') = \hat{C}(\tilde{S})$, т. е. длина реализуемой конфигурации совпадает с длиной реализующего её СОР. Задача первого этапа алгоритма $\mathcal{A}_\varepsilon$ — найти кратчайшую n-реализуемую конфигурацию. Из леммы 2 следует

Утверждение 1. *Длина кратчайшей n-реализуемой конфигурации Θ^* удовлетворяет оценке $C(\Theta^*) \leqslant C^*_{\max}(1 + 2\delta) + \kappa\delta + \kappa$.*

Для нахождения такой конфигурации проанализируем, как расписание работы J_k влияет на k-реализуемую конфигурацию.

Пусть $\tilde{S}$ — СОР для множества работ $\{J_1, \ldots, J_k\}$, и пусть работа J_k имеет большую операцию O_{qk}, которая в $\tilde{S}$ выполняется на некоторой машине $M_l \in \mathcal{M}_q$ в сегментах с номерами $\hat{s}, \ldots, \hat{c}$. Спрашивается, какие СМ-конфигурации может иметь машина M_l в этих сегментах? Легко понять, что набор таких СМ-конфигураций не может быть произвольным.

Пусть $\hat{s} < \hat{c}$ и в сегментах $\hat{s}$ и $\hat{c}$ машина M_l имеет СМ-конфигурации $\Omega_b = (\alpha_b, \beta_b, \gamma_b)$ и $\Omega_e = (\alpha_e, \beta_e, \gamma_e)$ соответственно. Тогда должны выполняться соотношения: $\gamma_b > 0$, $\alpha_e > 0$, $\gamma_b + \alpha_e + \nu(\hat{c} - \hat{s} - 1) = \tilde{p}_{qk}$, а в сегментах I_g с номерами $\hat{s} < g < \hat{c}$ машина M_l должна иметь конфигурацию Ω_T.

В случае $\hat{s} = \hat{c}$ должно выполняться неравенство $\beta_b \geqslant \tilde{p}_{qk}$.

Четверку натуральных чисел $B_q = (\hat{s}_q, \hat{c}_q; b_q, e_q)$ назовём *краткой конфигурацией операции* O_{qk}, если выполняются требования:
(a) $1 \leqslant \hat{s}_q \leqslant \hat{c}_q \leqslant \eta'$;
(b) $b_q, e_q \in \{1, \ldots, T\}$;
(c) если O_{qk} — большая операция, то при $\hat{s}_q < \hat{c}_q$ выполняются соотношения $\gamma_{b_q} > 0$; $\alpha_{e_q} > 0$; $\gamma_{b_q} + \alpha_{e_q} + \nu(\hat{c}_q - \hat{s}_q - 1) = \tilde{p}_{qk}$, а при $\hat{s}_q = \hat{c}_q$ — соотношения $\beta_{b_q} \geqslant \tilde{p}_{qk}$;
(d) если O_{qk} — малая операция, то $\hat{s}_q = \hat{c}_q$.

Набор $\bar{B}(J_k) = (B_1, \ldots, B_s)$ четвёрок $B_q = (\hat{s}_q, \hat{c}_q; b_q, e_q)$, $q = 1, \ldots, s$, удовлетворяющих требованиям (a)–(d), назовём *краткой конфигурацией работы* J_k, если выполняется требование

(e) $\hat{c}_q \leqslant \hat{s}_{q+1},\ q = 1, \ldots, s-1$.

Заметим, что при задании краткой конфигурации операции O_{qk} некоторые компоненты четверки B_q могут быть избыточными. Так, например, при $\hat{s}_q = \hat{c}_q$ не используется параметр e_q, а в случае, когда O_{qk} — малая операция, не используются ни b_q, ни e_q (их можно задавать произвольно).

По большой операции O_{qk} и номеру i СМ-конфигурации $\Omega_i = (\alpha_i, \beta_i, \gamma_i)$ определим СМ-конфигурации с номерами $\varphi(i), \psi(i)$ и $\omega_{qk}(i)$:

$$\Omega_{\varphi(i)} = (\alpha_i, \beta_i, 0);\ \Omega_{\psi(i)} = (0, \beta_i, \gamma_i);\ \Omega_{\omega_{qk}(i)} = (\alpha_i, \beta_i - \tilde{p}_{qk}, \gamma_i).$$

Заметим, что при $\gamma_i > 0,\ \alpha_i > 0,\ \tilde{p}_{qk} > 0$ соответственно выполняются неравенства

$$\varphi(i) < i,\ \psi(i) < i,\ \omega_{qk}(i) < i. \tag{11}$$

Если из расписания $\tilde{S}$ для работ $J_1, \ldots, J_k$ «изъять» все операции работы J_k, то конфигурация $\theta^{qg}(\tilde{S})$ каждого цеха $\mathcal{M}_q$ претерпит изменения в некоторых сегментах I_g. Конечно, изменения будут касаться лишь тех сегментов, в которых выполняется операция O_{qk}. Например, если большая операция O_{qk} имела краткую конфигурацию $B_q = (\hat{s}, \hat{c}; b, e)$, то при $\hat{s} < \hat{c}$ после изъятия из расписания операции O_{qk} в конфигурации $\theta^{q\hat{s}}$ на единицу уменьшится компонента $m_b^{q\hat{s}}$ (число машин, имеющих СМ-конфигурацию Ω_b) и на единицу увеличится компонента $m_{\varphi(b)}^{q\hat{s}}$; в конфигурации $\theta^{q\hat{c}}$ на единицу уменьшится компонента $m_e^{q\hat{c}}$ и на единицу увеличится компонента $m_{\psi(e)}^{q\hat{c}}$; в каждой из конфигураций $\theta^{qg},\ \hat{s} < g < \hat{c}$, на единицу уменьшится компонента m_T^{qg} и на единицу увеличится компонента m_1^{qg}.

Пусть задана краткая конфигурация $\bar{B}(J_k) = (B_1, \ldots, B_s)$ работы J_k, $B_q = (\hat{s}_q, \hat{c}_q; b_q,\ e_q)$. *Полную конфигурацию* $\Theta(J_k, \bar{B}(J_k))$ *работы* J_k определим как семейство $(T+1)$-мерных векторов $\{\theta^{qg}(J_k) \mid q = 1, \ldots, s;$ $g = 1, \ldots, \eta'\}$, где вектор $\theta^{qg}(J_k) = (\theta_1^{qg}, \ldots, \theta_T^{qg};\ \theta_{T+1}^{qg})$ определяется по краткой конфигурации $\bar{B}(J_k)$ следующим образом.

- Если O_{qk} — большая операция, то при $\hat{s}_q < \hat{c}_q$ полагаем $\theta_{b_q}^{q\hat{s}_q} := 1$, $\theta_{\varphi(b_q)}^{q\hat{s}_q} := -1$, $\theta_{e_q}^{q\hat{c}_q} := 1$; $\theta_{\psi(e_q)}^{q\hat{c}_q} := -1$; $\theta_T^{qg} := 1$, $\theta_1^{qg} := -1$ при любом $g,\ \hat{s}_q < g < \hat{c}_q$; при $\hat{s}_q = \hat{c}_q$ полагаем $\theta_{b_q}^{q\hat{s}_q} := 1,\ \theta_{\omega_{qk}(b_q)}^{q\hat{s}_q} := -1$. (Гарантия того, что мы не присваиваем одной и той же компоненте разные значения, следует из (11), требования (c) в определении краткой конфигурации работы и очевидного свойства, что большая операция имеет ненулевую длину.)

- Если O_{qk} — малая операция, то $\theta^{q\hat{s}_q}_{T+1} := p^*_{qk}$, где $p^*_{qk} = p'_{qk}/\delta^*$ — целое число.
- Все остальные компоненты векторов $\theta^{qg}(J_k)$, $q = 1, \ldots, s$; $g = 1, \ldots, \eta'$, равны нулю.

Пусть $\Theta' = \{\theta^{qg}\}$ — некоторая абстрактная конфигурация, $\Theta(J_k, \bar{B}(J_k)) = \{\theta^{qg}(J_k)\}$ — полная конфигурация работы J_k. Тогда *разность* $\Theta' \ominus \Theta(J_k, \bar{B}(J_k))$ определим как семейство векторов $\Theta'' = \{\theta^{qg} - \theta^{qg}(J_k)\}$.

Лемма 3. *Если все векторы $\theta^{qg} - \theta^{qg}(J_k)$ имеют неотрицательные компоненты, то Θ'' является абстрактной конфигурацией.*

Доказательство. Требуется доказать, что Θ'' удовлетворяет (8) и (9). Равенства (8) следуют из того, что Θ' удовлетворяет (8), и из определения векторов $\theta^{qg}(J_k)$. Что касается неравенств (9), то для малой операции O_{qk}, приписанной в конфигурации $\bar{B}(J_k)$ к сегменту $\hat{s}_q$, при вычитании вектора $\theta^{qg}(J_k)$ в неравенстве (9) изменяется лишь левая часть (L^{qg} уменьшается на p^*_{qk}), что, очевидно, сохраняет неравенство. Если же O_{qk} — большая операция, то для любого сегмента I_g, $\hat{s}_q \leqslant g \leqslant \hat{c}_q$, вычитание вектора $\theta^{qg}(J_k)$ из вектора θ^{qg} приводит к уменьшению на 1 некоторой компоненты m^{qg}_i и увеличению на 1 другой компоненты $m^{qg}_{i'}$, такой что $i' < i$ (из (11)) и $d_{i'} < d_i$, что приводит к уменьшению суммы $\sum\limits_{i=1}^{T} m^{qg}_i d_i$ в левой части неравенства (9) и неубыванию величины m'_{qg} в правой части неравенства (9). Таким образом, из справедливости (9) для конфигурации Θ' следует справедливость (9) для Θ''. Лемма 3 доказана.

Замечание 1. Пусть Θ' — k-реализуемая конфигурация (т. е. существует СОР $\tilde{S}$ для работ из $\mathcal{J}_k$ такое, что $\Theta(\tilde{S}) = \Theta'$). Тогда по расписанию $\tilde{S}$ однозначно определяются краткая $\bar{B}(J_k)$ и полная $\Theta(J_k, \bar{B}(J_k))$ конфигурации работы J_k. При этом разность $\Theta'' = \Theta' \ominus \Theta(J_k, \bar{B}(J_k))$ является $(k-1)$-реализуемой конфигурацией.

СОР $\tilde{S}'$ для работ из $\mathcal{J}_{k-1}$ получается удалением операций работы J_k из расписания $\tilde{S}$ и сдвигом влево оставшихся больших операций, выполнявшихся в $\tilde{S}$ непосредственно после удалённой большой операции O_{qk} работы J_k на той же машине в том же сегменте I_g — в случае, когда O_{qk} целиком выполняется в сегменте I_g.

Верно и обратное утверждение.

Лемма 4. *Пусть Θ' — абстрактная конфигурация и для работы J_k существует краткая конфигурация $\bar{B}(J_k)$ такая, что разность $\Theta'' = \Theta' \ominus \Theta(J_k, \bar{B}(J_k))$ является $(k-1)$-реализуемой конфигурацией.*

Тогда Θ' — k-реализуемая конфигурация.

Доказательство. Пусть $\tilde{S}''$ — СОР для множества работ $\{J_1, \ldots, J_{k-1}\}$, реализующее конфигурацию Θ''. Покажем, как из расписания $\tilde{S}''$ и конфигурации $\bar{B}(J_k)$ работы J_k получается СОР $\tilde{S}'$ для множества работ $\{J_1, \ldots, J_k\}$, реализующее конфигурацию Θ', т. е. как определяется расписание работы J_k и как, возможно, перестраивается при этом расписание работ $J_1, \ldots, J_{k-1}$.

Привязку операций работы J_k ко времени (т. е. указание номера $\tilde{s}(O_{qk})$ стартового κ'-интервала для каждой большой операции O_{qk} и номера $\hat{s}(O_{qk})$ сегмента привязки каждой малой операции) определяем согласно её краткой конфигурации $\bar{B}(J_k) = (B_1, \ldots, B_s)$; $B_q = (\hat{s}_q, \hat{c}_q; b_q, e_q)$, $q = 1, \ldots, s$:

для малой операции O_{qk} полагаем $\hat{s}(O_{qk}) := \hat{s}_q$;

для большой операции O_{qk} с $\hat{s}_q < \hat{c}_q$ полагаем $\tilde{s}(O_{qk}) := \nu\hat{s}_q + 1 - \gamma_{b_q}$;

для большой операции O_{qk} с $\hat{s}_q = \hat{c}_q$ полагаем $\tilde{s}(O_{qk}) := \nu(\hat{s}_q - 1) + 1 + \alpha_{b_q} + \beta_{b_q} - \tilde{p}_{qk}$.

Нетрудно убедиться, что при таком расписании компонента $\hat{c}_q$ конфигурации B_q совпадает с номером сегмента, в котором заканчивается операция O_{qk}. Таким образом, соблюдение требования ($\tilde{\mathrm{c}}$) из определения СОР вытекает из справедливости требования (e) для конфигурации $\bar{B}(J_k)$.

Далее нужно доопределить СОР $\tilde{S}$ для работы J_k, т. е. выполнить привязку больших операций O_{qk} к исполнителям-машинам с соблюдением требований СОР ($\tilde{\mathrm{a}}$), ($\tilde{\mathrm{b}}$), ($\tilde{\mathrm{d}}$) (или по крайней мере показать возможность осуществления такой привязки).

Пусть O_{qk} — большая операция. Если $\hat{s}_q = \hat{c}_q$, то операцию O_{qk} назначаем на любую машину цеха $\mathcal{M}_q$, имеющую в расписании $\tilde{S}''$ в сегменте $I_{\hat{s}_q}$ конфигурацию $\Omega_{\omega_{qk}(b_q)}$ (множество таких машин не пусто, так как в конфигурации $\Theta'' = \Theta(\tilde{S}'')$ выполнено неравенство $\theta^{q\hat{s}_q}_{\omega_{qk}(b_q)} \geqslant 1$).

Пусть теперь $\hat{s}_q < \hat{c}_q$. Назначаем операцию O_{qk} на любую машину M_i цеха $\mathcal{M}_q$, имеющую в сегменте $I_{\hat{s}_q}$ расписания $\tilde{S}''$ конфигурацию $\Omega_{\varphi(b_q)}$ (множество таких машин не пусто). Далее, если $\hat{s}_q + 1 < \hat{c}_q$, то нужно продолжить выполнение операции O_{qk} на машине, имеющей в сегменте $I_{\hat{s}_q+1}$ расписания $\tilde{S}''$ «пустую» конфигурацию Ω_1. Мы знаем, что такие машины есть, но не обязательно, что среди них имеется машина M_i, выбранная нами на предыдущем шаге. В то же время желательно продолжить выполнение операции O_{qk} *на той же машине* M_i, поскольку прерывания операций запрещены. Пусть это требование оказалось нарушенным, т. е. машина M_i имеет в сегменте $I_{\hat{s}_q+1}$расписания $\tilde{S}''$ конфигурацию $\Omega_i,\ i > 1$. При этом какая-то другая машина $M_{i'}$ $(i' \neq i)$ этого цеха в том же сегменте имеет конфигурацию Ω_1. Мы знаем, что в расписании $\tilde{S}''$ нет большой операции, которая бы выполнялась на машине M_i в сегменте $g \leqslant \hat{s}_q$ и продолжалась в сегменте $\hat{s}_q + 1$ (поскольку машина M_i в сегменте $\hat{s}_q$ имеет конфигурацию $\Omega_{\varphi(b_q)} = (\alpha, \beta, 0)$). Таким образом, все большие операции, выполняющиеся на M_i в сегменте $I_{\hat{s}_q+1}$, либо начинаются, либо полностью выполняются в этом сегменте. Это справедливо и по отношению к машине $M_{i'}$, что позволяет машинам M_i и $M_{i'}$ без каких-либо нарушений допустимости расписания $\tilde{S}''$ обменяться «клиентами», обслуживаемыми на данных машинах в сегментах с номерами $g = \hat{s}_q + 1, \ldots, \eta'$ (под «клиентами» мы понимаем большие операции). В результате такой трансформации расписания $\tilde{S}''$ машина M_i получает в сегменте $I_{\hat{s}_q+1}$ «пустую» конфигурацию Ω_1, что позволяет продолжить на ней операцию O_{qk}. Аналогично выполняем трансформацию расписания $\tilde{S}''$, если в каком-то из последующих сегментов $I_g,\ \hat{s}_q < g < \hat{c}_q$, машина M_i окажется в конфигурации, отличной от Ω_1, или если в сегменте $I_{\hat{c}_q}$ её конфигурация не совпадёт с $\Omega_{\psi(e_q)}$. Наконец, для постановки в расписание последней части операции O_{qk}, выполняемой в сегменте $I_{\hat{c}_q}$ и имеющей длительность α_{e_q} (κ'-интервалов), все большие операции O' машины M_i, входящие в «среднюю» часть расписания сегмента $I_{\hat{c}_q}$ (суммарной длительностью β_{e_q}), требуется сдвинуть вправо на величину α_{e_q}, т. е. увеличить на α_{e_q} величины $\tilde{s}(O')$ в записях $\tilde{S}(O')$ для этих операций.

Далее мы должны убедиться, что так построенное расписание $\tilde{S}'$ является СОР и его конфигурация $\Theta(\tilde{S}')$ совпадает с Θ'. Последнее очевидно. Что касается первого, то требования $(\tilde{a})$ и $(\tilde{b})$, касающиеся расписания больших операций, выполняются по построению расписания $\tilde{S}'$, а требование $(\tilde{d})$ следует из неравенств (9), выполняющихся для абстрактной конфигурации Θ'. Следовательно, Θ' является k-реализуемой конфигурацией. Лемма 4 доказана.

Из леммы 4 и замечания 1 вытекает

Лемма 5. *Абстрактная конфигурация Θ' является k-реализуемой $(k \geqslant 1)$, если и только если существует краткая конфигурация $\bar{B}(J_k)$ работы J_k такая, что разность $\Theta'' = \Theta' \ominus \Theta(J_k, \bar{B}(J_k))$ является $(k-1)$-реализуемой конфигурацией.*

Лемма 5 даёт теоретическую базу для применения метода динамического программирования на этапе 1 описываемого ниже алгоритма $\mathcal{A}_\varepsilon$.

Алгоритм состоит из трех этапов.

Этап 1. Нахождение всех k-реализуемых конфигураций $\Theta' \in \mathbf{\Theta}$ для $k = 0, 1, \ldots, n$. Нахождение СОР $\tilde{S}(\Theta')$ работ из $\mathcal{J}_k$ для каждой k-реализуемой конфигурации $\Theta' \in \boldsymbol{\Theta}$.

Этап 2. Нахождение кратчайшей n-реализуемой конфигурации Θ^*.

Этап 3. Преобразование СОР $\tilde{S}(\Theta^*)$ в допустимое расписание S исходной задачи.

Приведём более подробное описание каждого из этапов.

5. Алгоритм $\mathcal{A}_\varepsilon$

Этап 1. Определяются значения булевой функции $p(k, \Theta)$, $k \in \{0, 1, \ldots, n\}$, $\Theta \in \boldsymbol{\Theta}$, путём просмотра всех пар (k, Θ) в порядке неубывания k.

Шаг $k = 0$. $p(0, \Theta_\varnothing) := \mathbf{true}$; $p(0, \Theta') := \mathbf{false}$ при любом $\Theta' \in \boldsymbol{\Theta} \setminus \{\Theta_\varnothing\}$.

Шаг k $(k > 0)$. Для каждой конфигурации $\Theta' \in \boldsymbol{\Theta}$ вычисляется значение $p(k, \Theta')$ следующим образом. Сначала полагается $p(k, \Theta') := \mathbf{false}$, а затем это значение пересчитывается для каждой краткой конфигурации $\bar{B}(J_k) = (B_1, \ldots, B_s)$ работы J_k по формуле

$$p(k, \Theta') := p(k, \Theta') \vee p(k-1, \Theta' \ominus \Theta(J_k, \bar{B}(J_k))).$$

Ясно, что по окончании просмотра всех конфигураций $\bar{B}(J_k)$ функция $p(k, \Theta')$ будет иметь значение **true**, если и только если существует конфигурация $\bar{B}(J_k)$ такая, что $p(k-1, \Theta' \ominus \Theta(J_k, \bar{B}(J_k))) = \mathbf{true}$. С учётом леммы 5 индукцией по k нетрудно доказать, что $p(k, \Theta') = \mathbf{true}$, если и только если конфигурация Θ' является k-реализуемой.

При каждом истинном значении функции $p(k, \Theta')$ запоминается СОР $\tilde{S}(\Theta')$ работ из $\mathcal{J}_k$, реализующее конфигурацию Θ'. Расписание $\tilde{S}(\Theta')$ находится с помощью метода, описанного в доказательстве леммы 4. Вычисляется длина $\hat{C}(\tilde{S}(\Theta'))$ расписания $\tilde{S}$.

Этап 2. Находится кратчайшая n-реализуемая конфигурация Θ^*:

$$\Theta^* = \arg\min\{C(\Theta') \,|\, p(n, \Theta') = \mathbf{true}\}.$$

Этап 3. Пусть $\tilde{S}(\Theta^*)$ — СОР, построенное по оптимальной конфигурации Θ^*. Расписание $\tilde{S}(\Theta^*)$ делит все большие операции на две категории: *внутренние* операции, выполняющиеся целиком внутри одного сегмента (в краткой конфигурации $B_q = (\hat{s}_q, \hat{c}_q; b_q, e_q)$ такой операции O_{qk} выполняется $\hat{s}_q = \hat{c}_q$), и *переходящие* операции, выполняющиеся в нескольких сегментах (для них имеем $\hat{s}_q < \hat{c}_q$). Преобразование СОР $\tilde{S}(\Theta^*)$ в расписание S выполняется за три шага.

Шаг 1 (растяжение СОР). Каждый сегмент I_g (интервал длины κ) заменяется на *расширенный сегмент* I'_g — интервал длины $\kappa' = \kappa + 2\kappa\delta$. При этом на каждой машине M_i, $i = 1, \ldots, m$, в каждом расширенном сегменте I'_g, $g = 1, \ldots, \eta'$, под большие операции резервируются такие же по длительности интервалы: интервал длины $\kappa'\alpha^{ig}$ — в начале, длины $\kappa'\beta^{ig}$ — в середине и длины $\kappa'\gamma^{ig}$ — в конце сегмента I'_g, если машина M_i в расписании $\tilde{S}(\Theta^*)$ имела CM-конфигурацию $(\alpha^{ig}, \beta^{ig}, \gamma^{ig}) \neq (\nu, 0, 0)$ в сегменте I_g. Если же машина имела в сегменте I_g конфигурацию $(\nu, 0, 0)$, то весь сегмент I'_g резервируется под ту же переходящую большую операцию. Таким образом, по сравнению с СОР $\tilde{S}(\Theta^*)$ в новом СОР $\tilde{S}'$ под большие переходящие операции отводятся не меньшие, а под большие внутренние — такие же по длине интервалы.

Далее каждая малая операция приписывается в СОР $\tilde{S}'$ к расширенному сегменту I'_g с тем же номером g. Поскольку длина сегментов увеличилась на $2\kappa\delta$, а место, резервируемое под большие операции на каждой машине M_i в каждом сегменте I'_g, осталось тем же самым (по крайней мере, это касается всех таких машин с неполной конфигурацией $(\alpha^{ig}, \beta^{ig}, \gamma^{ig})$, что $\alpha^{ig} + \beta^{ig} + \gamma^{ig} < \nu$), то в каждом сегменте I_g с неполной конфигурацией длина связного отрезка, свободного для размещения малых операций, увеличилась на $2\kappa\delta$. Этот отрезок (будем называть его *свободным*) располагается в сегменте I'_g между начальным отрезком длины $(\alpha^{ig} + \beta^{ig})\kappa'$ и конечным отрезком длины $\gamma^{ig}\kappa'$.

Шаг 2 (размещение малых операций). Все малые операции цеха $\mathcal{M}_q$, приписанные к сегменту I'_g, размещаются по машинам цеха $\mathcal{M}_q$ и по времени в свободные отрезки сегмента I'_g. Размещение малых операций выполняется жадным алгоритмом: свободный отрезок на какой-либо машине $M_i \in \mathcal{M}_q$ в сегменте I'_g заполняется малыми операциями в произвольном порядке до тех пор, пока это возможно (т. е. пока хватает места для очередной операции и есть ещё не распределённые операции). Такой способ размещения позволяет заполнить свободный отрезок длины x малыми операциями не менее чем на длину $x - \kappa\delta$ (так как каждая малая операция имеет длину не более $\kappa\delta$). Так как на шаге 1 каждый свободный отрезок растягивается на величину $2\kappa\delta$, то жадный алгоритм позволяет разместить в сегменте I'_g малые операции цеха $\mathcal{M}_q$ суммарной длительностью не менее $X \doteq \sum\limits_{\{M_i \in \mathcal{M}_q \,|\, x_i > 0\}} (x_i + \kappa\delta)$, где x_i — длина свободного отрезка на машине M_i в сегменте I_g расписания $\tilde{S}(\Theta^*)$. В обозначениях формулы (4) имеем

$$\sum_{\{M_i \in \mathcal{M}_q \,|\, x_i > 0\}} x_i = \kappa m_q - \kappa' \sum_{M_i \in \mathcal{M}_q} P^i_{qg}, \qquad \sum_{\{M_i \in \mathcal{M}_q \,|\, x_i > 0\}} \kappa\delta = \kappa\delta \cdot m'_{qg}.$$

Поэтому из (4) следует неравенство $l_{qg} \leqslant \sum\limits_{\{M_i \in \mathcal{M}_q \,|\, x_i > 0\}} (x_i + \kappa\delta)$. Это неравенство означает, что суммарная длительность малых операций цеха $\mathcal{M}_q$, приписанных к сегменту I'_g, не превосходит величины X. Следовательно, все малые операции будут размещены в сегменте I'_g без пересечений между собой и с большими операциями.

Шаг 3 (поцеховой сдвиг расписания вправо). Полученное после шага 2 расписание S' по-прежнему является ослабленным: в нём ограничения предшествования между операциями, принадлежащими одной работе, выполняются с некоторой «ошибкой». Точнее говоря, для каждой работы J_k вместо неравенств

$$c_{qk} \leqslant s_{q+1,k},\ q = 1, \ldots, s-1, \tag{12}$$

выполняются неравенства

$$\hat{c}(O_{qk}) \leqslant \hat{s}(O_{q+1,k}),\ q = 1, \ldots, s-1, \tag{13}$$

где $\hat{s}(O_{qk})$ и $\hat{c}(O_{qk})$ — номера расширенных сегментов, в которых начинается и заканчивается операция O_{qk} в расписании S'. В случае, когда соотношение (13) выполняется как равенство, возможно нарушение требования (12). Чтобы исключить возможность таких нарушений, достаточно сдвинуть расписание всего цеха $\mathcal{M}_{q+1}$ (в котором выполняется $(q+1)$-я операция каждой работы) вправо на длину одного расширенного сегмента (т. е. на $\kappa + 2\kappa\delta$) по отношению к расписанию цеха $\mathcal{M}_q$, в котором выполняются операции O_{qk} всех работ $J_k,\ k = 1, \ldots, n$. Иначе говоря, расписание $S'(\mathcal{M}_q)$ цеха $\mathcal{M}_q$ сдвигается вправо на величину $(\kappa + 2\kappa\delta)(q-1),\ q = 1, \ldots, s$.

Алгоритм $\mathcal{A}_\varepsilon$ описан.

Расписание S, получаемое в конце шага 3, является полностью допустимым. Оценим длину расписания S.

На шаге 1 этапа 3 СОР $\tilde{S}(\Theta^*)$ был растянут в $1 + 2\delta$ раз, а на шаге 3 его длина увеличилась ещё на $\kappa(1+2\delta)(s-1)$. Таким образом,

$$C_{\max}(S) \leqslant C(\tilde{S}(\Theta^*))(1+2\delta) + \kappa(1+2\delta)(s-1).$$

С учётом (1), (2) и утверждения 1 получаем окончательную оценку:

$$\begin{aligned} C_{\max}(S) &\leqslant C^*_{\max}(1+2\delta)^2 + (\kappa\delta+\kappa)(1+2\delta) + \kappa(s-1)(1+2\delta) \\ &\leqslant C^*_{\max}(1+2\delta)^2 + 2s(\kappa\delta+\kappa s)(1+2\delta)C^*_{\max} \\ &= C^*_{\max}(1+4\delta+4\delta^2+\delta(\delta+s)(1+2\delta)) \\ &= C^*_{\max}(1+(s+4)\delta+(2s+5)\delta^2+2\delta^3) \leqslant C^*_{\max}(1+\varepsilon). \end{aligned} \tag{14}$$

Оценим временну́ю сложность алгоритма $\mathcal{A}_\varepsilon$.

Безусловно, определяющим для оценки временно́й сложности является этап 1, на котором вычисляются значения функции $p(k,\Theta')$ при всех $k=0,\dots,n$ и по всем конфигурациям $\Theta'\in\mathbf{\Theta}$. Для вычисления каждого значения $p(k,\Theta')$ перебираются всевозможные краткие конфигурации $\bar{B}(J_k)=(B_1,\dots,B_s)$ работы J_k. Поскольку число последних оценивается сверху константой $(\eta' T)^{2s}\leqslant\left(\frac{4s+1+1/\kappa}{5\delta^6}\right)^{2s}$ (при фиксированных δ,κ и s), то временна́я сложность вычисления полной таблицы значений $p(k,\Theta')$, $k=0,\dots,n$; $\Theta'\in\mathbf{\Theta}$, не превосходит $O(nW)$, где $W=|\mathbf{\Theta}|$. Оценим величину W.

Число Z различных векторов $(m_1^{qg},\dots,m_T^{qg})$, удовлетворяющих (8), равно числу упорядоченных разбиений числа m_q на T неотрицательных слагаемых. С учётом $m_q\leqslant n$ получаем

$$Z=\binom{m_q+T-1}{T-1}\leqslant\frac{(n+1)\cdots(n+T-1)}{(T-1)!}\leqslant\frac{e^{\frac{T^2}{2n}}}{(T-1)!}\cdot n^{T-1}\leqslant O(n^{T-1}).$$

С учётом (9) и неравенства $m'_{qg}\leqslant n$ оценим величину L^{qg}:

$$L^{qg}\leqslant\frac{\kappa(1+\delta)n}{\delta^*}=\frac{(1+\delta)s}{\delta}\,n^2\leqslant O(n^2).$$

Таким образом, число различных векторов θ^{qg}, удовлетворяющих (8) и (9), не превосходит $O(n^{T+1})$. С учётом (7) число W оценивается сверху величиной $W\leqslant\ O(n^{(T+1)\eta' s})\leqslant O(n^{(T+2)\eta' s-1})\leqslant O\left(n^{\frac{\eta' s}{5\delta^6}-1}\right)$. Подставляя значения η' и δ, оценим сверху показатель степени:

$$\frac{\eta' s}{5\delta^6}\leqslant\frac{s}{5\delta^6}\left(\frac{2s}{\delta}+4s+1\right)<\frac{2s(s+1)}{5\delta^7}=\frac{2s(s+1)(2s+5)^7}{5\varepsilon^7}<0{,}1(2s+5)^9\cdot t^7.$$

Таким образом, временна́я сложность этапа 1 (и всего алгоритма $\mathcal{A}_\varepsilon$) не превосходит $O\left(n^{0{,}1(2s+5)^9\cdot t^7}\right)$. Итак, нами доказана

Теорема. *Для многопроцессорной задачи* flow shop *с n работами и s цехами существует полиномиальная (при ограниченном s) аппроксимационная схема, позволяющая для любого $\varepsilon=1/t$, где t — положительное* *целое, построить* $(1+\varepsilon)$*-приближающий алгоритм* $\mathcal{A}_\varepsilon$ *временно́й сложности* $O\left(n^{0{,}1(2s+5)^9\cdot t^7}\right)$.

Мы не интересуемся тем, сколь велика величина полученной оценки временно́й сложности алгоритма $\mathcal{A}_\varepsilon$. Для нас имеет значение лишь характер теоретической зависимости этой величины от параметров n и ε. В этом смысле было бы интересно выяснить возможность построения PTAS для данной задачи, в оценке временно́й сложности которой в показателе степени при n отсутствовала бы зависимость от параметров s и $1/\varepsilon$ (или хотя бы от одного из них). Принципиальная возможность для этого существует, хотя полностью избежать экспоненциальной зависимости временно́й сложности от s, по-видимому, не удастся (ввиду уже упомянутого результата из [22]).

ЛИТЕРАТУРА

1. **Севастьянов С. В.** Некоторые обобщения задачи Джонсона // Управляемые системы. Сб. научн. тр. Вып. 21. Новосибирск: Институт математики СО АН СССР, 1981. С. 45–61.
2. **Севастьянов С. В.** Построение приближенного расписания для системы поточного типа // Управляемые системы. Сб. научн. тр. Вып. 31. Новосибирск: Институт математики СО АН СССР, 1993. С. 66–71.
3. **Buten R. E., Shen V. Y.** A scheduling model for computer systems with two classes of processors // Proc. of the 1973 Sagamore Comput. Conference Parallel Processing. 1973. P. 130–138.
4. **Chen B.** Scheduling multiprocessor flow shops // Advances in optimization and approximation. Dordrecht: Kluwer Acad. Publ., 1994. P. 1–8.
5. **Chen B.** Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at one stage // J. Oper. Res. Soc. 1995. V. 46, N 2. P. 234–244.
6. **Chen B., Potts C. N., Woeginger G. J.** A review of machine scheduling: complexity, algorithms and approximability // Handbook of Combinatorial Optimization. V. 3. Boston: Kluwer Acad. Publ., 1998. P. 21–169.
7. **Garey M. R., Johnson D. S.** Strong NP-completeness results: motivation, examples and implications // J. Assoc. Comput. Mach. 1978. V. 25, N 3. P. 499–508.
8. **Garey M. R., Johnson D. S., Sethi R.** The complexity of flowshop and jobshop scheduling // Math. Oper. Res. 1976. V. 1, N 2. P. 117–129.
9. **Gupta J. N. D.** Two-stage hybrid flowshop scheduling problem // J. Oper. Res. Soc. 1988. V. 39, N 4. P. 359–364.
10. **Hall L. A.** Approximability of flow shop scheduling // Math. Programming. 1998. Series B. V. 82, N 1–2. P. 175–190. scheduling // Math. Oper. Res. 1976. V. 1, N 2. P. 117–129.

11. **Hochbaum D. S., Shmoys D. B.** Using dual approximation algorithms for scheduling problems: theoretical and practical results // J. Assoc. Comput. Mach. 1987. V. 34, N 1. P. 144–162.

12. **Hoogeveen J. A., Lenstra J. K., Veltman B.** Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard // Eur. J. Oper. Res. 1996. V. 89, N 1–2. P. 172–175.

13. **Jansen K., Sviridenko M. I.** Polynomial time approximation schemes for the multiprocessor open and flow shop scheduling problem // 17th annual symposium on theoretical aspects of computer science, STACS 2000. Proc. Berlin: Springer, 2000. P. 455-565 (Lecture Notes in Comput. Sci.; V. 1770).

14. **Jansen K., Solis-Oba R., Sviridenko M. I.** Makespan minimization in job shops: a linear time approximation scheme // SIAM J. Discrete Math. 2003. V. 16, N 2. P. 288–300.

15. **Koulamas C., Kyparisis G. J.** Asymptotically optimal linear time algorithms for two-stage and three-stage flexible flow shops // Naval Res. Logist. 2000. V. 47, N 3. P. 259–268.

16. **Langston M. A.** Interstage transportation planning in the deterministic flow-shop environment // Oper. Res. 1987. V. 35, N 4. P. 556–564.

17. **Lee C. Y., Vairaktarakis G. L.** Minimizing makespan in hybrid flowshops // Oper. Res. Lett. 1994. V. 16, N 3. P. 149–158.

18. **Sahni S.** Algorithms for scheduling independent tasks // J. Assoc. Comput. Mach. 1976. V. 23, N 1. P. 116–127.

19. **Schuurman P., Woeginger G. J.** A polynomial time approximation scheme for the two-stage multiprocessor flow shop problem // Theoret. Comput. Sci. 2000. V. 237, N 1–2. P. 105–122.

20. **Sevastianov S. V.** Geometrical heuristics for multiprocessor flowshop scheduling with uniform machines at each stage // J. Sched. 2002. V. 5, N 3. P. 205–225.

21. **Sriskandarajah C., Sethi S. P.** Scheduling algorithms for flexible flow shops: worst and average case performance // European J. Oper. Res. 1989. V. 43, N 2. P. 143–160.

22. **Williamson D. P., Hall L. A., Hoogeveen J. A., Hurkens C. A. J., Lenstra J. K., Sevastianov S. V., Shmoys D. B.** Short shop schedules // Oper. Res. 1997. V. 45, N 2. P. 288–294.

Адрес автора:

Институт математики
им. С. Л. Соболева СО РАН,
пр. Академика Коптюга, 4,
630090 Новосибирск,
Россия.
E-mail: seva@math.nsc.ru

Статья поступила
27 июля 2006 г.

Переработанный вариант —
15 ноября 2006 г.