

НЕСТРОГОЕ СУММИРОВАНИЕ ВЕКТОРОВ НА ПЛОСКОСТИ И ЕГО ПРИМЕНЕНИЕ В ЗАДАЧАХ ТЕОРИИ РАСПИСАНИЙ*)

С. В. Севастьянов

На плоскости с заданной нормой рассматриваются конечные семейства векторов с нулевой суммой и не более чем единичной длиной каждого вектора. Исследуется возможность нахождения такого порядка суммирования векторов заданного семейства, при котором для любых двух последовательных частичных сумм хотя бы одна лежит в заданной области G (такое суммирование называем «нестрогим»). Доказывается, что при определенных соотношениях между областью G и единичным шаром нормы существует перестановка векторов, задающая нестрогое суммирование векторов в G , и эта перестановка находится за полиномиальное время. Применение полученной теоремы к четырем задачам на построение кратчайшего расписания выполнения n работ на m машинах позволяет для задачи open-shop ($m = 3$) расширить класс входов, допускающих эффективное точное решение, а для задач Акерса — Фридмана ($m = 3$), циклически замкнутых маршрутов ($m = 3$) и flow-shop ($m = 4$) построить эффективные алгоритмы нахождения расписаний с рекордными оценками точности.

Введение

В данной работе развивается тема применения нестрогого суммирования векторов (НСВ) в теории расписаний, начатая в [1]. Напомним, что в [1] рассмотрены следующие три задачи теории расписаний с n работами и m машинами. В первой задаче все работы проходят машины в одном и том же порядке $(1, 2, \dots, m)$ (известна как задача *flow-shop*). Во второй задаче все операции каждой работы, за исключением одной выделенной операции, могут выполняться независимо во времени, а выделенная операция («сборка») выполняется после завершения всех остальных операций работы (известна как задача *о сборочной линии*).

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 93-01-00489) и Международного научного фонда (грант NQC000).

Наконец, в третьей задаче, формулируемой для трех машин, все работы делятся на две группы, при этом работы первой группы проходят машины в порядке $(1, 2, 3)$, а работы второй группы — в порядке $(2, 1, 3)$. (Такую задачу мы называем здесь задачей R213.) Было показано сведение указанных задач к трем различным задачам НСВ в пространстве размерности $m - 1$, позволяющее по приближенному решению задачи НСВ находить приближенное решение соответствующей задачи теории расписаний. В случае плоскости для приближенного решения трех задач НСВ с гарантированными оценками точности предложен единый алгоритм полиномиальной трудоемкости. Этот алгоритм опирается на доказанный в [1] факт существования перестановки π , задающей нестрогое суммирование s -семейства векторов в неограниченной замкнутой области $G \subset R^2$ при условии, что всякая O -хорда области G имеет длину (в норме s) не меньше единицы. Наконец, применение алгоритма решения задач НСВ к трем рассматриваемым задачам теории расписаний с тремя машинами позволило сконструировать алгоритмы полиномиальной трудоемкости для построения расписаний с гарантированными оценками отклонения от оптимума; для двух из полученных оценок доказана их неувлучшаемость.

В настоящей работе будем придерживаться аналогичной схемы изложения результатов. Сначала в § 2 устанавливаются достаточные условия, при выполнении которых существует нестрогое суммирование заданного s -семейства векторов в области $G \subset R^2$. Применение этого результата к трем задачам НСВ на плоскости позволяет с полиномиальной трудоемкостью находить для них приближенные решения с гарантированными оценками точности. Затем в § 3, 4 рассматриваются четыре задачи на построение кратчайшего расписания выполнения n работ на m машинах: open-shop ($OS(m)$), flow-shop ($FS(m)$) при $m = 3, 4$, Акерса — Фридмана ($AF(m)$) при $m = 3$ и циклически замкнутых маршрутов (R231) при $m = 3$. Устанавливается связь между рассматриваемыми задачами и задачами о нестрогом суммировании векторов в пространствах размерности $m - 1$ или $m - 2$. Наконец, применение алгоритмов решения задач НСВ из § 2 к решению задач на построение расписаний позволяет, во-первых, расширить класс входов задачи $OS(3)$, допускающих эффективное точное решение, а во-вторых, — сконструировать алгоритмы нахождения расписаний с лучшими оценками отклонения от оптимума в задачах $FS(4)$, $AF(3)$, R231. Одним из следствий полученных результатов является построение алгоритма линейной от n трудоемкости решения задачи $FS(3)$ с неувлучшаемой оценкой точности.

§ 1. Предварительные сведения и постановки задач

Сначала сформулируем четыре задачи на построение расписания минимальной длины, а также введем необходимые понятия и обозначения. Далее будут сформулированы результаты, полученные для этих задач, и проведено их сравнение с известными результатами. Рассматриваемые здесь задачи на построение расписания удобно представлять как частные случаи следующей общей задачи.

Задача G. Имеется множество машин $\mathcal{M} = \{1, 2, \dots, m\}$ и множество работ $\mathcal{N} = \{1, 2, \dots, n\}$. Каждая работа j состоит из m операций $\{o_{j1}, \dots, o_{jm}\} = O^j$, причем известно, что операция o_{ji} выполняется на машине i , $1 \leq i \leq m$, за время t_{ji} . Для каждого $j \in \mathcal{N}$ порядок выполнения операций из множества O^j задан ориентированным графом $G_j = (O^j, U_j)$. При этом наличие дуги $(o_{ji}, o_{jk}) \in U_j$ означает, что операция o_{jk} не может начаться прежде, чем закончится операция o_{ji} . (Будем обозначать это также $o_{ji} \rightarrow o_{jk}$.) Требуется построить расписание $S = \{s_{ji} \geq 0 \mid j \in \mathcal{N}, i \in \mathcal{M}\}$ (т. е. для каждой операции o_{ji} указать момент ее начала s_{ji}), минимизирующее время завершения последней работы, т. е. величину $T(S) \doteq \max_{j,i} (s_{ji} + t_{ji})$ (традиционно именуемую «длиной расписания»), и удовлетворяющее следующим требованиям:

- каждая операция o_{ji} неразрывна, т. е. выполняется в течение всего интервала времени $I_{ji} \doteq (s_{ji}, s_{ji} + t_{ji})$;
- в каждый момент времени выполняется не более одной операции каждой работы, т. е.

$$I_{ji} \cap I_{jk} = \emptyset \quad \text{при любых } j \in \mathcal{N}; i, k \in \mathcal{M}, i \neq k;$$

- в каждый момент времени выполняется не более одной операции на каждой машине, т. е.

$$I_{ji} \cap I_{ki} = \emptyset \quad \text{при любых } i \in \mathcal{M}; j, k \in \mathcal{N}, j \neq k. \quad (1)$$

В рамках общей задачи G задачи OS, FS, AF и R231 различаются лишь множествами U_j , задающими частичные порядки на множествах O^j .

В задаче OS множество U_j пусто для любого $j \in \mathcal{N}$. Это означает, что операции работы j могут выполняться в любом порядке.

В задаче FS граф G_j представляет собой цепь $o_{j1} \rightarrow o_{j2} \rightarrow \dots \rightarrow o_{jm}$, $j \in \mathcal{N}$. Таким образом, для каждой работы задан маршрут прохождения машин из множества \mathcal{M} и этот маршрут $(1, 2, \dots, m)$ одинаков для всех работ.

В задаче АГ для каждой работы $j \in \mathcal{N}$ также задается маршрут (i_1^j, \dots, i_m^j) прохождения машин $\{1, 2, \dots, m\}$, причем каждая машина встречается один раз и для разных работ эти маршруты могут быть различными. (Возможно одновременное присутствие до $m!$ различных маршрутов.)

Наконец, задача R231 является сужением задачи АГ на случай трех машин и наличия в системе всего двух различных маршрутов: маршрута $(1, 2, 3)$ — для работ из множества \mathcal{N}_1 и маршрута $(2, 3, 1)$ — для работ из множества \mathcal{N}_2 ($\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2$).

Обозначим $M_i = \sum_{j=1}^n t_{ji}$, $M = \max_i M_i$, $K = \max_{j,i} t_{ji}$, S_{opt} — оптимальное расписание. В [2] установлено свойство оптимальных решений задачи OS, которое целесообразно назвать свойством *существования беспростойного оптимального расписания* (СБОР-свойство). Одновременно был описан довольно широкий подкласс эффективно решаемых задач OS. Как СБОР-свойство, так и указанный подкласс можно описать в терминах величин M и K . А именно, было показано, что если

$$M \geq \eta(m)K \quad (2)$$

для некоторой функции $\eta(m)$ от числа машин m , то

$$T(S_{\text{opt}}) = M, \quad (3)$$

т. е. в оптимальном расписании наиболее загруженная машина работает без простоев в течение всего интервала действия расписания. При этом оптимальное расписание находится за полиномиальное время.

Естественно возникает вопрос о нахождении наименьшей функции $\eta(m) = \eta^*(m)$ такой, что из (2) вытекает (3). Из алгоритма Гонзалеса — Сахни [3] следует, что $\eta^*(2) = 2$. В работах [4–6] были найдены верхние оценки функции $\eta^*(m)$ для произвольного m , улучшающие оценку из [2]. В [6] была приведена также нижняя оценка на $\eta^*(m)$. Эти оценки функции $\eta^*(m)$ при $m \geq 3$ можно записать следующим образом:

$$2m - 2 \leq \eta^*(m) \leq \min \left\{ m^2 - 1 + \frac{1}{m-1}, \frac{16}{3}m \log_2 m + \frac{61}{9}m - 7.4 \right\}.$$

В частности, из этих оценок при $m = 3$ получаем

$$4 \leq \eta^*(3) \leq 8.5. \quad (4)$$

Применяя технику НСВ из § 2, получаем лучшую верхнюю оценку для $\eta^*(3)$:

$$\eta^*(3) \leq 7.$$

Другим важным свойством задач типа job-shop и open-shop является так называемое свойство *локализации оптимумов*. Согласно этому

свойству оптимум любой индивидуальной задачи из класса X находится в интервале $I_X = [M, M + \mu_X(m)K]$, где $\mu_X(m)$ — некоторая функция от числа машин. Обозначим через $\mu_X^*(m)$ точную верхнюю границу множества значений функции $(T(S_{\text{opt}}) - M)/K$ по всем индивидуальным задачам из класса X с m машинами. В случае локализации оптимумов *перестановочных* расписаний задач из класса FS аналогичную функцию обозначим через $\tilde{\mu}(m)$. Минимальный интервал I_X , задаваемый функцией μ_X^* , будем обозначать I_X^* . Справедливы следующие соотношения, установленные в [6] и [1]:

$$m - 1 + \left\lfloor \frac{m - 1}{2} \right\rfloor \leq \tilde{\mu}(m) \leq m^2 - 3m + 3 + \frac{1}{m - 2}, \quad (5)$$

$$\tilde{\mu}(2) = \mu_{\text{FS}}^*(2) = 1, \quad \tilde{\mu}(3) = \mu_{\text{FS}}^*(3) = 3.$$

Из (5) при $m = 4$ получаем

$$4 \leq \tilde{\mu}(4) \leq 7.5.$$

В § 4 описано сведение задачи FS(m) к задаче НСВ в пространстве размерности $m - 2$. Это сведение позволяет улучшить верхнюю оценку значения $\tilde{\mu}(4)$ ($\tilde{\mu}(4) \leq 6$), а также построить алгоритм приближенного решения задачи FS(3), который находит перестановочное расписание S_π с неулучшаемой оценкой длины $(T(S_\pi) \leq M + 3K)$ за линейное от n время. Наконец, сведение задач AF(3) и R231 к задачам НСВ на плоскости и применение следствий теоремы 1 из § 2 позволяет вывести оценки

$$\mu_{\text{R231}}^* \leq 5, \quad \mu_{\text{AF}}^*(3) \leq 5.5.$$

(Последняя улучшает оценку $\mu_{\text{AF}}^*(3) \leq 6$, полученную в [7].) Поскольку для обеих функций известная нижняя оценка имеет вид $\mu_X^*(3) \geq 3$ ($X \in \{\text{R231}, \text{AF}\}$), задача нахождения точных значений $\mu_{\text{AF}}^*(3)$, μ_{R231}^* остается нерешенной.

Отметим также, что для каждой из рассматриваемых в § 4 задач $X \in \{\text{FS}, \text{AF}, \text{R231}\}$ доказательство улучшенной верхней оценки (μ_X) функции μ_X^* сопровождается эффективным построением приближенного расписания со значением длины из интервала I_X . Это гарантирует оценку абсолютного отклонения найденного решения от оптимума, не зависящую от числа работ.

§ 2. Нестрогое суммирование векторов в заданной области на плоскости

Введем несколько понятий и определений.

Пусть R^m — m -мерное векторное пространство над полем действительных чисел, а $\Sigma(X)$ — сумма векторов конечного семейства $X \subset R^m$.

Перестановку чисел от 1 до n будем называть просто *перестановкой*.

ОПРЕДЕЛЕНИЕ 1. Пусть в пространстве R^m задана норма s . Конечное семейство векторов $X = \{x_1, \dots, x_n\} \subset R^m$ называется s -семейством, если

$$\Sigma(X) = 0, \quad (6)$$

а норма каждого вектора не превосходит единицы.

ОПРЕДЕЛЕНИЕ 2. Пусть в R^m заданы область G и семейство векторов $X = \{x_1, \dots, x_n\}$ с нулевой суммой. Будем говорить, что перестановка $\pi = (\pi_1, \dots, \pi_n)$ задает *нестрогое суммирование векторов* из X в области G , если при любом $k = 1, \dots, n$ из $x_{\pi}^{k-1} \notin G$ следует $x_{\pi}^k \in G$, где $x_{\pi}^k \doteq \sum_{j=1}^k x_{\pi_j}$.

Будем говорить, что перестановка π задает *нестрогое суммирование векторов* из $X \subset R^m$ в семействе областей $\mathcal{P} = \{G_i \mid i = 1, \dots, \nu\}$, и обозначать этот факт $(X, \pi) \in S_m \mathcal{P}$, если π задает нестрогое суммирование в каждой из областей $G_i \in \mathcal{P}$, $i = 1, \dots, \nu$.

Для любых областей $G_i \subset R^m$ справедливо

ЗАМЕЧАНИЕ 1. Перестановка π , задающая нестрогое суммирование векторов из X в пересечении областей $G = \bigcap_{i=1}^k G_i$, задает и нестрогое суммирование этих векторов в семействе областей $\{G_i \mid i = 1, \dots, k\}$.

Для любого вектора $a = (a(1), \dots, a(m)) \in R^m$, $a \neq 0$, и числа $\beta \in R$ определим полупространство $P(a, \beta) = \{x \in R^m \mid (a, x) \leq \beta\}$, где

$$(a, x) = \sum_{i=1}^m a(i)x(i).$$

Через e_i обозначим i -й орт пространства R^m . В R^m зададим норму \hat{s} , единичный шар которой (с центром в нуле) определяется соотношением

$$\hat{H}_m = \{x \in R^m \mid |x(i_1)| \leq 1, |x(i_1) - x(i_2)| \leq 1 \text{ при всех } i_1, i_2 = 1, \dots, m\}.$$

По-другому, шар \hat{H}_m может быть определен следующим образом.

Лемма 1. $\hat{H}_m = \text{conv}(B \cup (-B)) \doteq B'$, где $B = [0, 1]^m$.

ДОКАЗАТЕЛЬСТВО. Очевидно включение $B \subseteq \hat{H}_m$. Так как \hat{H}_m симметрично, то $-B \subseteq \hat{H}_m$. Так как \hat{H}_m выпукло, то $B' = \text{conv}(B \cup (-B)) \subseteq \hat{H}_m$. Докажем обратное включение.

Пусть $z \in \hat{H}_m$. Докажем, что $z \in B'$. Обозначим $z_{\min} = \min_i z(i)$, $z_{\max} = \max_i z(i)$, $\bar{e} = (1, 1, \dots, 1) \in R^m$. Тогда $-1 \leq z_{\min} \leq z_{\max} \leq 1$ и $z_{\max} - z_{\min} \leq 1$.

Если $z_{\min} \geq 0$ или $z_{\max} \leq 0$, то $z \in B$ или $z \in -B$ соответственно, откуда в обоих случаях имеем $z \in B'$. Пусть $z_{\min} < 0 < z_{\max}$. Определим

векторы $z' = z - z_{\max}\bar{e}$, $z'' = z - z_{\min}\bar{e}$. Тогда для любого $i = 1, \dots, m$ имеем $z'(i) = z(i) - z_{\max} \in [-1, 0]$ и $z''(i) = z(i) - z_{\min} \in [0, 1]$. Следовательно, $z' \in -B$ и $z'' \in B$. Таким образом,

$$\begin{aligned} B' = \text{conv}(B \cup (-B)) &\ni \frac{-z_{\min}}{z_{\max} - z_{\min}} z' + \frac{z_{\max}}{z_{\max} - z_{\min}} z'' \\ &= \left(\frac{-z_{\min}}{z_{\max} - z_{\min}} + \frac{z_{\max}}{z_{\max} - z_{\min}} \right) z + \left(\frac{z_{\max} z_{\min}}{z_{\max} - z_{\min}} - \frac{z_{\max} z_{\min}}{z_{\max} - z_{\min}} \right) \bar{e} = z. \end{aligned}$$

Лемма 1 доказана.

Сформулируем четыре задачи (НСВ i , $i = 1, \dots, 4$) о нестрогом суммировании векторов в семействах полупространств пространства R^m . Отметим, что под такими же номерами $i = 1, 2, 3$ в [1] фигурируют другие задачи НСВ i . Однако это обстоятельство не должно вводить читателя в заблуждение, поскольку мы не будем обращаться к задачам из [1].

Задача НСВ1(m). Для заданного \hat{s} -семейства векторов $X = \{x_1, \dots, x_n\} \subset R^m$ требуется найти перестановку $\pi = (\pi_1, \dots, \pi_n)$ и набор вещественных чисел $b_{m+1} = (\beta_1, \dots, \beta_{m+1})$ такие, что выполнено $(X, \pi) \in S_m \mathcal{P}_1(m, b_{m+1})$ для $\mathcal{P}_1(m, b_{m+1}) = \{P(e_1 - e_2, \beta_1), \dots, P(e_{m-1} - e_m, \beta_{m-1}), P(e_m, \beta_m), P(-e_1, \beta_{m+1})\}$, а целевая функция

$$\theta_1(b_{m+1}) = \sum_{i=1}^{m+1} \beta_i$$

принимает минимальное возможное значение.

Задача НСВ2(m). Для заданного \hat{s} -семейства векторов $X = \{x_1, \dots, x_n\} \subset R^m$ требуется найти перестановку $\pi = (\pi_1, \dots, \pi_n)$ и набор вещественных чисел $b_m = (\beta_1, \dots, \beta_m)$ такие, что выполнено $(X, \pi) \in S_m \mathcal{P}_2(m, b_m)$ для $\mathcal{P}_2(m, b_m) = \{P(e_1 - e_2, \beta_1), \dots, P(e_{m-1} - e_m, \beta_{m-1}), P(e_m, \beta_m)\}$, а целевая функция $\theta_2(b_m) = \sum_{i=1}^m \beta_i$ принимает минимальное возможное значение.

Пусть $N_6 \doteq \{1, \dots, 6\}$. На множестве N_6 введем циклический порядок $1 \rightarrow 2 \rightarrow \dots \rightarrow 6 \rightarrow 1$, определив для каждого элемента $i \in N_6$ его предыдущий ($p(i)$) и последующий ($s(i)$) элементы. Через I обозначим множество «несоседних» пар из N_6 , т. е.

$$I \doteq \{(i, j) \mid i, j \in N_6; i \neq j; i \neq s(j); i \neq p(j)\}.$$

Задача НСВ3. Для заданного \hat{s} -семейства векторов $X = \{x_1, \dots, x_n\} \subset R^2$ и векторов $a_1 = e_1$, $a_2 = e_2$, $a_3 = e_2 - e_1$, $a_4 = -e_1$, $a_5 = -e_2$, $a_6 = e_1 - e_2$ требуется найти перестановку $\pi = (\pi_1, \dots, \pi_n)$ и набор вещественных чисел $b_6 = (\beta_1, \dots, \beta_6)$ такие, что выполнено

$(X, \pi) \in S_2 \mathcal{P}_3(2, b_6)$ для $\mathcal{P}_3(2, b_6) = \{P(a_1, \beta_1), \dots, P(a_6, \beta_6)\}$, а целевая функция

$$\theta_3(b_6) = \max_{(i,j) \in I} (\beta_i + \beta_j)$$

принимает минимальное возможное значение.

Задача НСВ4. Для заданного \hat{s} -семейства векторов $X = \{x_1, \dots, x_n\} \subset R^2$ требуется найти перестановку $\pi = (\pi_1, \dots, \pi_n)$ и набор вещественных чисел $b_3 = (\beta_1, \beta_2, \beta_3)$ такие, что выполнено $(X, \pi) \in S_2 \mathcal{P}_1(2, b_3)$, а функционал

$$\theta_4(b_3) = \max\{\beta_1 + \beta_2, \beta_2 + \beta_3\}$$

принимает минимальное возможное значение.

Докажем лемму, утверждающую возможность сведения задачи НСВ2 в пространстве размерности m к задаче НСВ1 в пространстве размерности $m - 1$.

Лемма 2. Пусть имеется алгоритм \mathcal{A} , который для любых натуральных m и n для всякого \hat{s} -семейства векторов $X = \{x_1, \dots, x_n\} \subset R^m$ с трудоемкостью $\mathcal{T}_{\mathcal{A}}(m, n)$ находит приближенное решение задачи НСВ1(m) с целевой функцией $\theta_1(b_{m+1}) \leq \theta_1^*(m)$. Тогда существует алгоритм \mathcal{A}' трудоемкости $O(\mathcal{T}_{\mathcal{A}}(m - 1, n))$, который для задачи НСВ2(m) находит решение с целевой функцией

$$\theta_2(b_m) \leq \theta_1^*(m - 1).$$

Доказательство. Пусть для всякого m мы умеем решать задачу НСВ1(m) с оценкой

$$\theta_1(b_{m+1}) \leq \theta_1^*(m),$$

и пусть на входе задачи НСВ2(m) имеется \hat{s} -семейство векторов $X = \{x_1, \dots, x_n\} \subset R^m$. Опишем алгоритм \mathcal{A}' решения задачи НСВ2(m), состоящий из трех шагов.

Алгоритм \mathcal{A}'

1. Для каждого $j = 1, \dots, n$ спроецируем вектор $x_j \in X \subset \hat{H}_m$ параллельно вектору e_1 на гиперплоскость $\Gamma = \{x \in R^m \mid x(1) = 0\}$ (для этого достаточно занулить первую координату вектора x_j) и полученную проекцию обозначим x'_j . Из определения шара \hat{H}_m следует, что $X' \doteq \{x'_1, \dots, x'_n\} \subset \hat{H}_{m-1}$, т. е. в пространстве R^{m-1} координат $\{2, \dots, m\}$ все векторы x'_j имеют не более чем единичную длину в норме \hat{s} . И так как наследуется свойство $\sum_{j=1}^n x'_j = 0$, то семейство X' является \hat{s} -семейством в пространстве R^{m-1} .

2. Относительно входа X' найдем приближенное решение задачи НСВ1($m-1$), т. е. такую перестановку $\pi^* = (\pi_1^*, \dots, \pi_n^*)$ и такой набор вещественных чисел $b_m = (\beta_2, \dots, \beta_m, \beta_1)$, что

$$(X', \pi^*) \in S_{m-1} \mathcal{P}', \quad (7)$$

$$\theta_1(b_m) = \sum_{i=1}^m \beta_i \leq \theta_1^*(m-1), \quad (8)$$

где $\mathcal{P}' = \{P'(e_2 - e_3, \beta_2), P'(e_3 - e_4, \beta_3), \dots, P'(e_{m-1} - e_m, \beta_{m-1}), P'(e_m, \beta_m), P'(-e_2, \beta_1)\}$.

3. Пусть $y'_\nu = \sum_{j=1}^\nu x'_{\pi_j}$ — ν -я вершина траектории суммирования

векторов $x'_j \in X'$ согласно перестановке π^* , а $y_\nu = \sum_{j=1}^\nu x_{\pi_j}$ — аналогичная вершина при суммировании векторов $x_j \in X$. Найдем k такое, что $y_k(1) = \max_{\nu=1, \dots, n} y_\nu(1)$, и в качестве искомой перестановки $\pi = (\pi_1, \dots, \pi_n)$ возьмем циклический сдвиг перестановки π^* на k элементов:

$$\pi_j = \begin{cases} \pi_{j+k}^* & \text{при } j+k \leq n, \\ \pi_{j+k-n}^* & \text{при } j+k > n. \end{cases}$$

Алгоритм \mathcal{A}' описан.

Каждому полупространству $P'(a, \beta) \subset \Gamma$ при $a \in \Gamma$ поставим в соответствие полупространство $P(a, \beta) \subset R^m$ и через \mathcal{P} обозначим семейство полупространств $P \subset R^m$, соответствующих полупространствам $P' \in \mathcal{P}'$. Поскольку каждый вектор x'_j отличается от x_j на вектор, коллинеарный орту e_1 , имеем $y_\nu - y'_\nu = \alpha_\nu e_1$ для некоторого $\alpha_\nu \in R$. Отсюда для любого $a \in \Gamma$ получаем $(y_\nu, a) = (y'_\nu, a) + \alpha_\nu(e_1, a) = (y'_\nu, a)$. Если $y'_\nu \in P'(a_i, \beta_i) \in \mathcal{P}'$, то $y_\nu \in P(a_i, \beta_i) \in \mathcal{P}$. Таким образом, (7) влечет

$$(X, \pi^*) \in S_m \mathcal{P}, \quad (9)$$

т. е. нестрогую суммируемость векторов исходного семейства X в семействе полупространств \mathcal{P} .

Обозначим $z_\nu = \sum_{j=1}^\nu x_{\pi_j}$. Тогда с учетом $\sum_{j=1}^n x_j = 0$ имеем

$$z_\nu = \begin{cases} y_{k+\nu} - y_k & \text{при } k+\nu \leq n, \\ y_{k+\nu-n} - y_k & \text{при } k+\nu > n. \end{cases}$$

Отсюда $z_\nu(1) \leq 0$, или

$$z_\nu \in P(e_1, 0), \quad \nu = 1, \dots, n. \quad (10)$$

Относительно семейства X циклический сдвиг перестановки π^* на k элементов равносильен переносу начала координат в k -ю вершину траектории суммирования векторов $\{x_j\}$ (т. е. в точку y_k). При этом траектория суммирования совпадает с исходной траекторией с точностью до

параллельного переноса. В результате свойство нестрогой суммируемости векторов X выполнится (при суммировании согласно перестановке π) относительно полупространств семейства $\mathcal{P} + y \doteq \{P + y \mid P \in \mathcal{P}\}$, где $y \doteq -y_k$, т. е.

$$(X, \pi) \in S_m\{\mathcal{P} + y\}.$$

Покажем, что смещение всех полупространств семейства \mathcal{P} на один и тот же вектор y не меняет значения целевой функции θ_1 . Действительно, если $x \in P(a_i, \beta_i) + y$, то $(a_i, x - y) \leq \beta_i$, или $(a_i, x) \leq \beta_i + (a_i, y) \doteq \beta'_i$. Таким образом, $P(a_i, \beta_i) + y = P(a_i, \beta'_i)$, где

$$\begin{aligned} \sum \beta'_i &= \sum \beta_i + \left(\sum_{i=1}^m a_i, y \right) \\ &= \sum \beta_i + ((e_2 - e_3) + (e_3 - e_4) + \dots + (e_{m-1} - e_m) + e_m - e_2, y) = \sum \beta_i. \end{aligned}$$

Наконец, из (10) следует, что если $z_\nu \in P(-e_2, \beta_1)$, то

$$z_\nu \in P(e_1, 0) \cap P(-e_2, \beta_1) \subset P(e_1 - e_2, \beta_1).$$

Это означает, что перестановка π , обеспечивающая нестрогое суммирование векторов X в полупространстве $P(-e_2, \beta_1) \in \mathcal{P} + y$ и удовлетворяющая (10), обеспечивает и нестрогую суммируемость векторов X в полупространстве $P(e_1 - e_2, \beta_1)$. Заменяя в семействе $\mathcal{P} + y$ полупространство $P(-e_2, \beta_1)$ полупространством $P(e_1 - e_2, \beta_1)$, получаем семейство полупространств задачи НСВ2(m) со значением функционала

$$\theta_2(b_m) = \sum_{i=1}^m \beta_i \leq \theta_1^*(m - 1).$$

Трудоемкость алгоритма \mathcal{A}' складывается из его трудоемкости на шагах 1, 2, 3 и не превосходит $O(1) + \mathcal{T}_{\mathcal{A}}(m - 1, n) + O(n)$. Поскольку $\mathcal{T}_{\mathcal{A}}(m - 1, n) = \Omega(n)$ (такое минимальное число действий необходимо для того, чтобы записать искомую перестановку π), получаем оценку, декларируемую в лемме. Лемма 2 доказана.

Далее в этом параграфе будем рассматривать двумерное векторное пространство R^2 . Для любого вектора $x \in R^2$, $x \neq 0$, вводим обозначения: $l(x) = \{tx \mid t \in R\}$ — прямая и $Q(x) = \{tx \mid t \in R^+\}$ — луч в направлении вектора x ; $L(x)$, $R(x)$ — замкнутые полуплоскости, лежащие слева и справа от прямой $l(x)$ при движении по лучу $Q(x)$ в бесконечность.

Для $x, y \in R^2$ через $[x, y]$ обозначим замкнутый отрезок, соединяющий точки x и y , т. е. $[x, y] = \{\lambda x + (1 - \lambda)y \mid \lambda \in [0, 1]\}$.

ОПРЕДЕЛЕНИЕ 3. Будем говорить, что три вектора $y_1, y_2, y_3 \in R^2$ образуют *нормальную тройку*, если существуют числа $\lambda_i \geq 0, i = 1, 2, 3$, такие, что

$$\sum \lambda_i y_i = 0, \quad \sum \lambda_i > 0.$$

Легко видеть, что три вектора на плоскости образуют нормальную тройку, если и только если они не лежат в одной открытой полуплоскости.

Пусть в R^2 определена некоторая норма s , и H — ее единичный шар с центром в нуле, т. е. $H = \{x \in R^2 \mid \|x\| \leq 1\}$. Величину $\|x\|$ будем называть *длиной* вектора x . Кроме того, для каждого вектора x определим его угловую координату $u(x) \in [0, 2\pi)$. Для определенности будем считать, что угол возрастает при повороте вектора против часовой стрелки (как это принято изображать на комплексной плоскости). Считаем также, что $u(0) = 0$.

Для любых двух векторов a и b таких, что $u(a) \neq u(b), a \neq 0, b \neq 0$, через $C(a, b)$ обозначим сектор плоскости от луча $Q(a)$ против часовой стрелки до луча $Q(b)$. Более формально

$$C(a, b) = \begin{cases} L(a) \cap R(b), & \text{если } b \in L(a), \\ L(a) \cup R(b), & \text{если } b \in R(a). \end{cases}$$

В случае $u(a) = u(b)$ полагаем

$$C(a, b) = C(b, a) = Q(a) = Q(b).$$

Через $C^0(a, b)$ обозначим открытый сектор $C(a, b) \setminus (Q(a) \cup Q(b))$, а через $\angle C(a, b)$ — величину угла сектора $C(a, b)$, т. е.

$$\angle C(a, b) = \begin{cases} u(b) - u(a), & \text{если } u(b) \geq u(a), \\ u(b) - u(a) + 2\pi, & \text{если } u(b) < u(a). \end{cases}$$

Теорема 1. Пусть на плоскости заданы норма s с единичным шаром H и выпуклое множество G такое, что

$$\text{если } x_1 \notin G, x_2 \notin G \text{ и } [x_1, x_2] \cap H \neq \emptyset, \text{ то } \|x_2 - x_1\| > 1. \quad (11)$$

Тогда для любого s -семейства векторов $X = \{x_1, \dots, x_n\} \subset R^2$ существует перестановка $\pi = (\pi_1, \dots, \pi_n)$, задающая нестрогое суммирование векторов X в G . Искомая перестановка может быть найдена алгоритмом \mathcal{A}^* трудоемкости $O(n \log n)$.

ДОКАЗАТЕЛЬСТВО. Заметим, что из свойства (11) и выпуклости G следует, что $H \subset G$ и всякая хорда множества G , пересекающая H , имеет длину не меньше 1. Наоборот, из двух последних свойств с добавлением условия замкнутости G вытекает (11). Мы, однако, не будем использовать замкнутость G при доказательстве теоремы.

Опишем алгоритм \mathcal{A}^* нахождения искомой перестановки π .

Алгоритм \mathcal{A}^*

Этап 1 (нумерация и построение дерева поиска).

Нумеруем векторы $x_i \in X$ по возрастанию их угловой координаты $u(x_i)$, при равенстве угловых координат — по возрастанию длины, наконец, при равенстве обеих координат — по возрастанию исходных номеров векторов. Таким образом, все нулевые векторы получают наименьшие номера (до k_0 включительно, где $k_0 \geq 0$). Трудоемкость алгоритма нумерации — $O(n \log n)$ операций.

Далее согласно Тарьяну [8] строим сбалансированное по глубине двоичное дерево поиска (T_0) на множестве вершин $I_0 = \{1, \dots, n\}$ по ключу $u(x_i)$, $i \in I_0$. Напомним, что такое дерево поиска обладает следующим полезным свойством: для каждой внутренней (не концевой) вершины i дерева T_0 все вершины $\{j\}$ в ее левом поддереве имеют ключи $u(x_j) \leq u(x_i)$, а все вершины в правом поддереве — ключи $u(x_j) \geq u(x_i)$. «Сбалансированность по глубине» означает, что расстояние от корня дерева до любого из его листьев не превосходит $O(\log n)$.

При построении такого дерева для каждой вершины i определим ее ранг $r(i)$ как максимальную степень двойки, на которую делится число i . Таким образом, ранги принимают значения $0, 1, \dots, \lfloor \log_2 n \rfloor$, причем максимальный ранг $r = \lfloor \log_2 n \rfloor$ достигается на единственной вершине $i_0 = 2^r$, которую принимаем за корень дерева T_0 . Для каждой вершины $i = 1, \dots, n$, отличной от корня i_0 , в качестве ее родителя $p(i)$ в дереве T_0 выбираем одну из двух вершин $i + 2^{r(i)}$, $i - 2^{r(i)}$. Выбор осуществляется следующим образом:

если $i + 2^{r(i)} > n$, то $p(i) := i - 2^{r(i)}$;

иначе если $r(i + 2^{r(i)}) = r(i) + 1$, то $p(i) := i + 2^{r(i)}$;

иначе $p(i) := i - 2^{r(i)}$.

Так как в любом случае имеем $r(p(i)) \geq r(i) + 1$, то глубина дерева не превосходит максимального ранга, т. е. $\lfloor \log_2 n \rfloor$. (Предоставляем читателю убедиться, что так определенное дерево является двоичным, т. е. у каждой вершины может быть не более двух детей.)

Этап 2 (построение перестановки π).

Перестановку π строим по шагам $k = 1, \dots, n$. После шага k известны первые k значений π_1, \dots, π_k перестановки π и множество I_k номеров векторов, еще не включенных в частичную сумму $\sigma_k \doteq \sum_{i=1}^k x_{\pi_i}$.

Известно также дерево поиска T_k с множеством вершин I_k , причем глубина дерева не превосходит величины r .

Для $k = 1, \dots, k_0$ полагаем $\pi_k = k$, т. е. первыми в искомом порядке суммирования идут нулевые векторы. Исключив номера $\{1, \dots, k_0\}$ из

множества I_0 , получим множество I_{k_0} , и удалив соответствующие вершины из дерева T_0 , получим дерево T_{k_0} . (Удаление любой вершины из n -вершинного дерева поиска требует $O(\log n)$ операций [8, с. 46].)

Пусть выполнено k шагов этапа 2; $k_0 \leq k \leq n - 2$.

Шаг $k + 1$. Находим номера $p_k, s_k \in I_k$ векторов $x_i \in X_k$, имеющих ближайшие к $u = u(-\sigma_k)$ значения угловой координаты (снизу и сверху соответственно). Пусть $I_k^-(u) \doteq \{i \in I_k \mid u(x_i) \leq u\}$, $I_k^+(u) \doteq \{i \in I_k \mid u(x_i) \geq u\}$. Положим

$$p_k = \begin{cases} \max\{j \in I_k^-(u)\}, & \text{если } I_k^-(u) \neq \emptyset, \\ \max\{j \in I_k\} & \text{в противном случае;} \end{cases}$$

$$s_k = \begin{cases} \min\{j \in I_k^+(u) \setminus \{p_k\}\}, & \text{если } I_k^+(u) \setminus \{p_k\} \neq \emptyset, \\ \min\{j \in I_k\} & \text{в противном случае.} \end{cases}$$

(Вместо p_k, s_k и σ_k там, где это не вызывает недоразумений, будем для краткости писать p, s и σ .) Нетрудно предложить алгоритм поиска вершин p_k, s_k в дереве поиска T_k , требующий $O(\log n)$ действий.

В качестве значения π_{k+1} будем брать один из двух номеров $\{p, s\}$. (Подробное правило выбора описано ниже.) Исключив (за $O(\log n)$ действий) вершину π_{k+1} из дерева T_k , приходим к дереву T_{k+1} .

Алгоритм описан.

Утверждение теоремы вытекает из трех лемм, доказываемых ниже.

Лемма 3. На каждом шаге $k = 1, \dots, n$ алгоритма \mathcal{A}^* тройка векторов $\langle \sigma, x_p, x_s \rangle$ является нормальной.

Доказательство. По определению векторов x_p, x_s на каждом шаге k имеем

$$-\sigma_k \in C(x_p, x_s), \quad (12)$$

$$X_k \subset C(x_s, x_p). \quad (13)$$

Отсюда $\angle C(x_s, x_p) \geq \pi$, или

$$\angle C(x_p, x_s) \leq \pi. \quad (14)$$

В противном случае при $\sigma_k \in C(x_s, x_p)$ с учетом (13) это противоречило бы равенству (6), а при $\sigma_k \in C(x_p, x_s)$ с учетом (12) получаем, что σ_k и все векторы из X_k лежат в одной полуплоскости, причем в предположении $\angle C(x_s, x_p) < \pi$ векторы x_p и x_s неколлинеарны, что также противоречит равенству (6).

Из (12) и (14) вытекает нормальность тройки $\langle \sigma, x_p, x_s \rangle$. Лемма 3 доказана.

Лемма 4. Пусть $\lambda_\sigma, \lambda_p, \lambda_s$ — коэффициенты, удостоверяющие нормальность тройки $\langle \sigma, x_p, x_s \rangle$, т. е.

$$\begin{aligned} \lambda_\sigma \sigma + \lambda_p x_p + \lambda_s x_s &= 0; \\ \lambda_i &\geq 0 \quad \text{при } i \in \{\sigma, p, s\}; \quad \sum \lambda_i > 0. \end{aligned}$$

Пусть также

$$\sigma + x_p \notin H \quad \text{и} \quad \sigma + x_s \notin H. \quad (15)$$

Тогда

$$\lambda_\sigma < \max\{\lambda_p, \lambda_s\}. \quad (16)$$

ДОКАЗАТЕЛЬСТВО. Пусть $\lambda_\sigma \geq \max\{\lambda_p, \lambda_s\}$. Тогда $\lambda_\sigma > 0$.

В случае $\lambda_p \geq \lambda_s$ имеем

$$\sigma + x_p = \left(1 - \frac{\lambda_p}{\lambda_\sigma}\right) x_p + \frac{\lambda_s}{\lambda_\sigma} (-x_s) = \gamma_p x_p + \gamma_s (-x_s),$$

где $\gamma_p \geq 0, \gamma_s \geq 0, \gamma_p + \gamma_s \leq 1, x_p \in H, -x_s \in H$ (в силу симметричности H). Следовательно, $\sigma + x_p \in \text{conv}\{x_p, -x_s, 0\} \subset H$, что противоречит (15).

В случае $\lambda_p \leq \lambda_s$ аналогично имеем

$$\sigma + x_s = \frac{\lambda_p}{\lambda_\sigma} (-x_p) + \left(1 - \frac{\lambda_s}{\lambda_\sigma}\right) x_s \in H,$$

что противоречит (15). Лемма 4 доказана.

Теперь мы имеем возможность более подробно описать работу алгоритма \mathcal{A}^* на шаге $k+1$.

Шаг $k+1$.

- Если $\sigma + x_p \in H$, то $\pi_{k+1} := p$.
- Если $\sigma + x_p \notin H$ и $\sigma + x_s \in H$, то $\pi_{k+1} := s$.
- Если $\sigma + x_p \notin H$ и $\sigma + x_s \notin H$, то находим неотрицательные коэффициенты $\lambda_\sigma, \lambda_p, \lambda_s$, удостоверяющие нормальность тройки $\langle \sigma, x_p, x_s \rangle$. По лемме 4 выполнено (16). Поэтому возможны лишь следующие три варианта соотношений между коэффициентами $\lambda_\sigma, \lambda_p, \lambda_s$.

Вариант 1 ($\lambda_s \leq \lambda_\sigma < \lambda_p$): $\pi_{k+1} := p$.

Вариант 2 ($\lambda_p \leq \lambda_\sigma < \lambda_s$): $\pi_{k+1} := s$.

Вариант 3 ($\lambda_\sigma < \min\{\lambda_p, \lambda_s\}$):

А) если $\sigma + x_p \in G$, то $\pi_{k+1} := p$;

В) если $\sigma + x_p \notin G, \sigma + x_s \in G$, то $\pi_{k+1} := s$;

С) если $\sigma + x_p \notin G, \sigma + x_s \notin G$, то $\pi_{k+1} := p, \pi_{k+2} := s$.

Шаг $k+1$ алгоритма \mathcal{A}^* описан.

Как видно из описания шага $k+1$, в случае С варианта 3 мы выполняем сразу два шага алгоритма.

Лемма 5. Пусть на шаге $t = k$ ($k_0 \leq k \leq n - 2$) выполнены соотношения

$$\sigma_t \in G, \quad (17)$$

$$(\sigma_t + Q(-x')) \cap H \neq \emptyset, \quad (18)$$

где x' — один из двух векторов x_p, x_s . Тогда (17), (18) выполняются либо на шаге $t = k + 1$, либо на шаге $t = k + 2$.

Доказательство. Заметим, что с учетом (6) равенство в (14) достигается лишь в том случае, если σ_k и все векторы из X_k коллинеарны. Поскольку доказательство утверждения леммы 5 в этом случае тривиально (в качестве вектора x_π на каждом шаге $t = k + 1, \dots, n$ выбирается вектор x_p , противоположно направленный с σ), далее будем предполагать, что

$$\angle C(x_p, x_s) < \pi. \quad (19)$$

Если $\sigma + x_p \in H$ или $\sigma + x_s \in H$, то полагаем соответственно $\pi_{k+1} := p$ или $\pi_{k+1} := s$, и соотношения (17), (18) выполнены при $t = k + 1$. Далее считаем, что имеет место соотношение (15). Для определенности будем также считать, что (18) при $t = k$ выполняется для $x' = x_s$, т. е.

$$(\sigma_t + Q(-x_s)) \cap H \neq \emptyset. \quad (20)$$

(В случае $x' = x_p$ доказательство полностью аналогично.) Поскольку выполнено (15), согласно описанию алгоритма находим коэффициенты $\lambda_\sigma, \lambda_p, \lambda_s$, удостоверяющие нормальность тройки $\langle \sigma, x_p, x_s \rangle$. Замечаем, что из (19) следует $\lambda_\sigma > 0$. Кроме того, $\lambda_p \neq 0$, так как в противном случае вектор x_s противоположно направлен вектору σ , и ввиду (20) $\sigma \in H$, $\sigma + x_s \in H$, что противоречит (15). При $\lambda_s = 0$ вектор x_p противоположно направлен σ , и для суммы $\sigma_{k+1} = \sigma + x_p$ соотношения (17), (18) выполнены при $t = k + 1$. Поэтому далее считаем, что

$$\lambda_i > 0 \quad \text{при} \quad i \in \{\sigma, p, s\}. \quad (21)$$

Рассмотрим три возможных варианта соотношений между коэффициентами $\lambda_\sigma, \lambda_p, \lambda_s$.

ВАРИАНТ 1 ($\lambda_s \leq \lambda_\sigma < \lambda_p$).

Имеем $\pi_{k+1} = p$, $\sigma_{k+1} = \sigma + x_p$. С учетом (19), (21) получаем

$$-\sigma_{k+1} = -\sigma - x_p = \left(\frac{\lambda_p}{\lambda_\sigma} - 1 \right) x_p + \frac{\lambda_s}{\lambda_\sigma} x_s \in C^0(x_p, x_s), \quad (22)$$

откуда $x_{s_{k+1}} = x_s$. Кроме того, имеем

$$y_1 \doteq -\frac{\lambda_s}{\lambda_\sigma} x_s \in H, \quad (23)$$

$$\sigma_{k+1} = \sigma + x_p = \sigma - \frac{\lambda_\sigma}{\lambda_p} \sigma - \frac{\lambda_s}{\lambda_p} x_s = \left(1 - \frac{\lambda_\sigma}{\lambda_p} \right) \sigma + \frac{\lambda_\sigma}{\lambda_p} y_1. \quad (24)$$

Из (23) и (24) вытекает соотношение (17) для $t = k + 1$. Докажем (18).

Из (20) следует существование точки $z = \sigma - \gamma x_s \in H$, где $\gamma \geq 0$. Обозначив $z_1 = \lambda_s z / (\lambda_\sigma \gamma + \lambda_s)$, будем иметь $z_1 \in H$ и

$$\begin{aligned} z_1 &= \frac{\lambda_s}{\lambda_\sigma \gamma + \lambda_s} (\sigma - \gamma x_s) = \frac{\lambda_s}{\lambda_\sigma \gamma + \lambda_s} \sigma + \frac{\lambda_\sigma \gamma}{\lambda_\sigma \gamma + \lambda_s} y_1 \\ &= (1 - \lambda') \sigma + \lambda' y_1 \in [y_1, \sigma], \end{aligned} \quad (25)$$

где $\lambda' \doteq \lambda_\sigma \gamma / (\lambda_\sigma \gamma + \lambda_s)$. Если $\lambda_\sigma / \lambda_p \geq \lambda'$, то из (24) и (23) имеем $\sigma_{k+1} \in [y_1, z_1] \subset H$, что противоречит условию (15). Таким образом, $\lambda_\sigma / \lambda_p < \lambda' = \lambda_\sigma \gamma / (\lambda_\sigma \gamma + \lambda_s)$, откуда $\gamma - (\lambda_\sigma / \lambda_p) \gamma - \lambda_s / \lambda_p \doteq \alpha > 0$. Но тогда

$$\begin{aligned} \sigma_{k+1} - \alpha x_{s_{k+1}} &= \sigma_{k+1} - \alpha x_s = \sigma - \frac{\lambda_\sigma}{\lambda_p} \sigma - \frac{\lambda_s}{\lambda_p} x_s - \left(\gamma - \frac{\lambda_\sigma}{\lambda_p} \gamma - \frac{\lambda_s}{\lambda_p} \right) x_s \\ &= \left(1 - \frac{\lambda_\sigma}{\lambda_p} \right) (\sigma - \gamma x_s) = \left(1 - \frac{\lambda_\sigma}{\lambda_p} \right) z \in H \cap (\sigma_{k+1} + Q(-x_{s_{k+1}})), \end{aligned}$$

что доказывает соотношение (18) при $t = k + 1$.

ВАРИАНТ 2 ($\lambda_p \leq \lambda_\sigma < \lambda_s$).

Заметим, что ввиду (20) этот вариант несимметричен варианту 1.

Полагаем $\pi_{k+1} = s$, $\sigma_{k+1} = \sigma + x_s$. Пусть $z = \sigma - \gamma x_s$ ($\gamma \geq 0$) — точка из H , существование которой вытекает из (20). Тогда

$$\sigma_{k+1} = \sigma + x_s \in \left[\sigma - \gamma x_s, \sigma + \frac{\lambda_s}{\lambda_\sigma} x_s \right] = \left[\sigma - \gamma x_s, -\frac{\lambda_p}{\lambda_\sigma} x_p \right] \subset H,$$

откуда вытекают соотношения (17), (18) при $t = k + 1$.

ВАРИАНТ 3 ($\lambda_\sigma < \min\{\lambda_p, \lambda_s\}$).

СЛУЧАЙ А ($\sigma + x_p \in G$).

Имеем $\pi_{k+1} = p$, $\sigma_{k+1} = \sigma + x_p \in G$, т. е. соотношение (17) при $t = k + 1$ выполнено. Для доказательства (18) определим векторы y_1 , z , z_1 и величины α , γ , λ' , как в варианте 1. Тогда согласно (22) имеем $-\sigma_{k+1} \in C^0(x_p, x_s)$, откуда $x_{s_{k+1}} = x_s$ и $x_{p_{k+1}} \in C(-x_s, x_p)$, или

$$-x_{p_{k+1}} \in C(x_s, -x_p). \quad (26)$$

Если $\lambda_\sigma / \lambda_p \geq \lambda'$, то из (24) и (25) имеем

$$z_1 \in [\sigma_{k+1}, \sigma] \subset \sigma_{k+1} + Q(-x_p),$$

откуда

$$(\sigma_{k+1} + Q(-x_p)) \cap H \neq \emptyset. \quad (27)$$

Кроме того, $\alpha \leq 0$ и

$$\sigma_{k+1} - \alpha x_s = \left(1 - \frac{\lambda_\sigma}{\lambda_p}\right) z \in H \cap (\sigma_{k+1} + Q(x_s)). \quad (28)$$

Из (26)–(28) с учетом (19) получаем

$$(\sigma_{k+1} + Q(-x_{p_{k+1}})) \cap H \neq \emptyset,$$

т. е. (18) выполнено для $t = k + 1$.

Если же $\lambda_\sigma/\lambda_p < \lambda'$, то $\alpha > 0$ и

$$\sigma_{k+1} - \alpha x_{s_{k+1}} = \sigma_{k+1} - \alpha x_s = \left(1 - \frac{\lambda_\sigma}{\lambda_p}\right) z \in H \cap (\sigma_{k+1} + Q(-x_{s_{k+1}})),$$

т. е. (18) выполнено при $t = k + 1$.

СЛУЧАЙ В ($\sigma + x_s \in G$).

Имеем $\pi_{k+1} = s$, $\sigma_{k+1} = \sigma + x_s \in G$, т. е. (17) выполнено при $t = k + 1$. Докажем (18).

Ввиду (19), (21) имеем

$$-\sigma_{k+1} = \frac{\lambda_p}{\lambda_\sigma} x_p + \left(\frac{\lambda_s}{\lambda_\sigma} - 1\right) x_s \in C^0(x_p, x_s),$$

откуда с учетом (19)

$$\begin{aligned} \angle C(-x_s, -\sigma_{k+1}) &< \pi, \\ x_{s_{k+1}} &\in C(x_s, -x_p) \subset C(x_s, \sigma_{k+1}), \end{aligned} \quad (29)$$

или

$$-x_{s_{k+1}} \in C(-x_s, -\sigma_{k+1}). \quad (30)$$

Так как $(\sigma_{k+1} + Q(-\sigma_{k+1})) \cap H \neq \emptyset$ и ввиду (20)

$$(\sigma_{k+1} + Q(-x_s)) \cap H \supset (\sigma + Q(-x_s)) \cap H \neq \emptyset,$$

с учетом (30) и (29) получаем $(\sigma_{k+1} + Q(-x_{s_{k+1}})) \cap H \neq \emptyset$, что доказывает наследование свойства (18) при $t = k + 1$.

СЛУЧАЙ С ($\sigma + x_p \notin G$, $\sigma + x_s \notin G$).

Имеем $\pi_{k+1} = p$, $\pi_{k+2} = s$, $\sigma_{k+2} = \sigma + x_p + x_s$ и $\sigma_{k+2} \in C^0(x_p, x_s)$, откуда с учетом (19) получаем $x_{p_{k+2}} \in C(-x_s, x_p)$, $x_{s_{k+2}} \in C(x_s, -x_p)$,

или

$$-x_{p_{k+2}} \in C(x_s, -x_p), \quad (31)$$

$$-x_{s_{k+2}} \in C(-x_s, x_p). \quad (32)$$

Определив z_1, y_1, λ' и α , как в варианте 1, замечаем, что случай $\sigma_{k+1} \in [z_1, \sigma]$ невозможен (вытекало бы, что $\sigma_{k+1} \in G$). Таким образом, $z_1 \in (\sigma_{k+1}, \sigma]$, откуда $\lambda' < \lambda_\sigma / \lambda_p$ и $\alpha < 0$.

Рассмотрим случаи $\alpha \geq -1$ и $\alpha < -1$.

СЛУЧАЙ 1: $\alpha \in [-1, 0)$. Поскольку $\sigma_{k+1} - \alpha x_s \in [\sigma_{k+1}, \sigma_{k+2}]$ и $\sigma_{k+1} \notin G$, с учетом (11) и $\|\sigma_{k+2} - \sigma_{k+1}\| = \|x_s\| \leq 1$ получаем (17) для $t = k + 2$.

Покажем, что при $t = k + 2$ имеет место (18) для $x' = x_{s_{k+2}}$. Действительно,

$$\sigma_{k+2} - (1 + \alpha)x_s = \sigma_{k+1} - \alpha x_s = \left(1 - \frac{\lambda_\sigma}{\lambda_p}\right) z \in (\sigma_{k+2} + Q(-x_s)) \cap H. \quad (33)$$

Кроме того,

$$\begin{aligned} \sigma_{k+2} + \frac{\lambda_p(1 + \alpha)}{\lambda_s + \lambda_\sigma \gamma} x_p &= \sigma + x_s + \frac{\lambda_s + \lambda_\sigma \gamma + \lambda_p + \lambda_p \alpha}{\lambda_s + \lambda_\sigma \gamma} x_p \\ &= \sigma + x_s - \frac{\lambda_p + \lambda_p \gamma}{\lambda_s + \lambda_\sigma \gamma} \frac{\lambda_\sigma}{\lambda_p} \sigma - \frac{\lambda_p + \lambda_p \gamma}{\lambda_s + \lambda_\sigma \gamma} \frac{\lambda_s}{\lambda_p} x_s \\ &= \frac{\lambda_s + \lambda_\sigma \gamma - \lambda_\sigma - \lambda_\sigma \gamma}{\lambda_s + \lambda_\sigma \gamma} \sigma + \frac{\lambda_s + \lambda_\sigma \gamma - \lambda_s - \lambda_s \gamma}{\lambda_s + \lambda_\sigma \gamma} x_s \\ &= \frac{\lambda_s - \lambda_\sigma}{\lambda_s + \lambda_\sigma \gamma} \sigma - \frac{\lambda_s - \lambda_\sigma}{\lambda_s + \lambda_\sigma \gamma} \gamma x_s = \frac{\lambda_s - \lambda_\sigma}{\lambda_s + \lambda_\sigma \gamma} (\sigma - \gamma x_s) \\ &= \frac{\lambda_s - \lambda_\sigma}{\lambda_s + \lambda_\sigma \gamma} z \doteq \gamma' z \doteq z_2, \end{aligned} \quad (34)$$

где $\gamma' \in (0, 1)$. Поэтому

$$(\sigma_{k+2} + Q(x_p)) \cap H \neq \emptyset. \quad (35)$$

Из (32), (33), (35) с учетом (19) получаем

$$(\sigma_{k+2} + Q(-x_{s_{k+2}})) \cap H \neq \emptyset,$$

т. е. (18) выполняется при $t = k + 2$.

СЛУЧАЙ 2: $\alpha < -1$. Из (34) следует, что $z_2 = \sigma_{k+2} + \lambda_2 x_p \in H$, где $\lambda_2 = \lambda_p(1 + \alpha)/(\lambda_s + \lambda_\sigma \gamma) \in (-1, 0)$. Так как $[\sigma + x_s, \sigma_{k+2}] \cap H \neq \emptyset$, то с учетом соотношений $\|\sigma_{k+2} - \sigma - x_s\| = \|x_p\| \leq 1$, $\sigma + x_s \notin G$ и (11) получаем $\sigma_{k+2} \in G$, т. е. (17) выполнено для $t = k + 2$. Покажем, что при $t = k + 2$ имеет место (18) для $x' = x_{p_{k+2}}$.

Так как $z_2 \in \sigma_{k+2} + Q(-x_p)$, то

$$(\sigma_{k+2} + Q(-x_p)) \cap H \neq \emptyset. \quad (36)$$

Кроме того, из (33) следует, что

$$\sigma_{k+2} - (1 + \alpha)x_s = \left(1 - \frac{\lambda_\sigma}{\lambda_p}\right) z \in (\sigma_{k+2} + Q(x_s)) \cap H. \quad (37)$$

Из (31), (36), (37) с учетом (19) получаем

$$(\sigma_{k+2} + Q(-x_{p_{k+2}})) \cap H \neq \emptyset,$$

т. е. (18) выполняется при $t = k + 2$. Лемма 5 доказана.

Вернемся к доказательству теоремы 1.

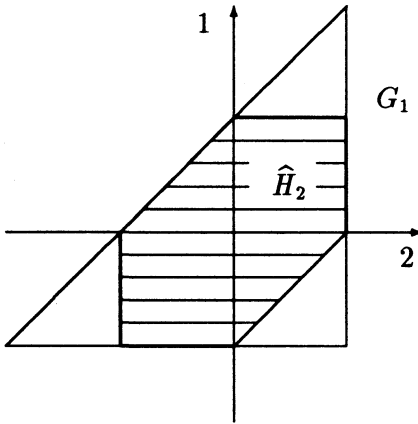
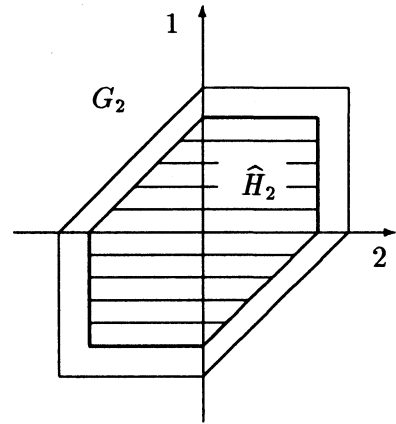
Ясно, что трудоемкость алгоритма на этапе 1 не превосходит $O(n \log n)$, поскольку как нумерация векторов, так и построение исходного дерева поиска T_0 могут быть выполнены с такой трудоемкостью. С этой же трудоемкостью мы реализуем этап 2, поскольку на каждом шаге $k = 1, \dots, n$ алгоритма построения перестановки π поиск вершин p_k, s_k и последующее удаление одной из них из дерева T_k могут быть выполнены за $O(\log n)$ действий. При этом предполагаем, что проверка включений $x \in H$ и $x \in G$, осуществляемая на каждом шаге этапа 2 для векторов $x = \sigma + x_p, x = \sigma + x_s$, может быть выполнена за константу действий. Это безусловно верно в тех частных случаях применения теоремы 1, которые будут рассмотрены в следствиях 1–3. Однако сформулированная в теореме оценка трудоемкости останется справедливой и в том случае, если каждая такая проверка потребует $O(\log n)$ действий.

Для завершения доказательства теоремы 1 остается показать, что получаемая алгоритмом \mathcal{A}^* перестановка π задает нестрогое суммирование векторов X в области G . Указанное свойство перестановки легко доказывается индукцией по шагам $t = 0, 1, \dots, n$. Базис индукции основывается на соотношениях $\sigma_t = 0 \in H \subset G$ ($t = 0, 1, \dots, k_0$), из которых следуют соотношения (17), (18) при $t = 0, 1, \dots, k_0$, а индукционный шаг обеспечивается леммой 5. Теорема 1 доказана.

Приведем следствия из теоремы 1, позволяющие эффективно находить приближенные решения задач НСВ i , $i = 1, 3, 4$, в пространстве размерности 2, а также, с учетом леммы 2, решение задачи НСВ2 в пространстве размерности 3.

Рассмотрим пересечение G_1 полупространств семейства $\{P(a_i, \beta_i)\}$ в задаче НСВ1(m) (НСВ4) при $m = 2$ и $\beta_1 = \beta_2 = \beta_3 = 1$. Нетрудно заметить, что G_1 является треугольником, описанным вокруг единичного шара \hat{H}_2 (рис. 1).

Для задачи НСВ3 пересечение G_2 полупространств $P(a_i, \beta_i) \in \mathcal{P}_3(2, b_6)$ при $\beta_i = 5/4, i = 1, \dots, 6$, является шаром радиуса $5/4$ (рис. 2). Поскольку в обоих случаях имеем $\hat{H}_2 \subset G_i$ и всякая хорда выпуклого множества G_i , пересекающая \hat{H}_2 , имеет длину (в норме \hat{s}) не меньше 1, условие (11) теоремы 1 выполнено. Это обеспечивает возможность нестрогое суммирование векторов всякого \hat{s} -семейства X в каждом из множеств G_1, G_2 , причем искомая перестановка векторов находится эффективно. Отсюда, согласно замечанию 1, вытекает нестрогая суммируемость векторов из X в соответствующих семействах полуплоскостей.

Рис. 1. Область G_1 Рис. 2. Область G_2

При этом целевые функции задач НСВ1(2), НСВ3 и НСВ4 принимают значения

$$\theta_1(b_3) = 3, \theta_3(b_6) = 2.5, \theta_4(b_3) = 2.$$

Таким образом, сформулируем следствия из теоремы 1.

Следствие 1. Существует алгоритм трудоемкости $O(n \log n)$, который для всякого входа задачи НСВ1(2) находит ее приближенное решение такое, что

$$\theta_1(b_3) \leq 3.$$

Следствие 2. Существует алгоритм трудоемкости $O(n \log n)$, который для всякого входа задачи НСВ3 находит ее приближенное решение такое, что

$$\theta_3(b_6) \leq 2.5.$$

Следствие 3. Существует алгоритм трудоемкости $O(n \log n)$, который для всякого входа задачи НСВ4 находит ее приближенное решение такое, что

$$\theta_4(b_3) \leq 2.$$

§ 3. Задача open-shop для трех машин

В этом параграфе установлена зависимость между оптимальным решением задачи OS с m машинами и приближенным решением задачи НСВ1 ($m - 1$). В случае $m = 3$ использование этой зависимости и следствия 1 позволяет обнаружить более широкий подкласс задач open-shop для трех машин, допускающих их эффективное точное решение.

Теорема 2. Пусть имеется алгоритм \mathcal{A} , который при любых натуральных n и m для любого \hat{s} -семейства $\{x_1, \dots, x_n\} \subset R^m$ с трудоемкостью $\mathcal{T}_{\mathcal{A}}(m, n)$ решает задачу НСВ1(m) с гарантированной оценкой

$$\theta_1(b_{m+1}) \leq \theta_1^*(m). \quad (38)$$

Тогда существует алгоритм \mathcal{A}_1 , который по всякому входу задачи орешор с m машинами и n работами, удовлетворяющему условию

$$M \geq (\theta_1^*(m-1) + 2m - 2)K, \quad (39)$$

с трудоемкостью $O(\mathcal{T}_{\mathcal{A}}(m-1, n) + mn)$ находит оптимальное расписание длины M .

Доказательство. Пусть $\pi = (\pi_1, \dots, \pi_n)$ — некоторая перестановка чисел $1, 2, \dots, n$. Определим величины

$$\Delta_i \doteq \max_{k=1, \dots, n} \left(\sum_{j=1}^k t_{\pi_j, i} - \sum_{j=1}^{k-1} t_{\pi_j, i+1} \right), \quad i = 1, \dots, m, \quad (40)$$

где $t_{\pi_j, m+1} = t_{\pi_j, 1}$. Определим семейство векторов $\{d_1, \dots, d_n\} = D \subset R^{m-1}$, где

$$d_j(i) = t_{j, i} - t_{j, m}, \quad i = 1, \dots, m-1. \quad (41)$$

Тогда поскольку

$$\begin{aligned} \sum_{j=1}^k t_{\pi_j, i} - \sum_{j=1}^{k-1} t_{\pi_j, i+1} &= \sum_{j=1}^{k-1} (t_{\pi_j, i} - t_{\pi_j, i+1}) + t_{\pi_k, i} = \sum_{j=1}^k (t_{\pi_j, i} - t_{\pi_j, i+1}) + t_{\pi_k, i+1} \\ &\leq K + \min \left\{ \sum_{j=1}^{k-1} (t_{\pi_j, i} - t_{\pi_j, i+1}), \sum_{j=1}^k (t_{\pi_j, i} - t_{\pi_j, i+1}) \right\} \end{aligned} \quad (42)$$

и выполняются соотношения

$$t_{j, i} - t_{j, i+1} = \begin{cases} d_j(i) - d_j(i+1) & \text{при } i = 1, \dots, m-2, \\ d_j(m-1) & \text{при } i = m-1, \\ -d_j(1) & \text{при } i = m, \end{cases} \quad (43)$$

из (40), (42) и (43) получаем оценки

$$\Delta_i \leq K + \max_{k=1, \dots, n} \min \{d_{\pi}^{k-1}(i) - d_{\pi}^{k-1}(i+1), d_{\pi}^k(i) - d_{\pi}^k(i+1)\}, \quad (44)$$

$$i = 1, \dots, m-2,$$

$$\Delta_{m-1} \leq K + \max_{k=1, \dots, n} \min \{d_{\pi}^{k-1}(m-1), d_{\pi}^k(m-1)\}, \quad (45)$$

$$\Delta_m \leq K + \max_{k=1, \dots, n} \min \{-d_{\pi}^{k-1}(1), -d_{\pi}^k(1)\}, \quad (46)$$

где $d_{\pi}^k \doteq \sum_{j=1}^k d_{\pi_j}$.

Приведем алгоритм \mathcal{A}_1 построения оптимального расписания задачи OS при выполнении (39) и в условиях теоремы 2.

Алгоритм \mathcal{A}_1

Шаг 1. Сохраняя неизменными величины M и K , увеличим длительности отдельных операций так, чтобы выполнялись равенства

$$M_i = M, \quad i = 1, \dots, m. \quad (47)$$

Трудоемкость шага равна $O(mn)$. Чтобы не менять обозначения всех величин, зависящих от величин $\{t_{ji}\}$, будем предполагать, что равенства (47) выполнялись для исходной матрицы (t_{ji}) .

Из (47), (41) и определения нормы \hat{s} следует, что если величину K взять за новую единицу измерения времени, то семейство векторов D становится δ -семейством в пространстве R^{m-1} .

Шаг 2. Применив к семейству D алгоритм \mathcal{A} , найдем перестановку $\pi = (\pi_1, \dots, \pi_n)$, задающую нестрогое суммирование векторов из D в семействе полупространств $\mathcal{P}_1(m-1, b_m K)$, где $b_m = (\beta_1, \dots, \beta_m)$,

$$\sum_{i=1}^m \beta_i \leq \theta_1^*(m-1). \quad (48)$$

Из (44)–(46) следуют неравенства $\Delta_i \leq (1 + \beta_i)K$, откуда с учетом (48) получаем

$$\sum_{i=1}^m \Delta_i \leq (m + \theta_1^*(m-1))K. \quad (49)$$

Трудоемкость шага 2 совпадает с трудоемкостью алгоритма \mathcal{A} , т. е. равна $O(\mathcal{T}_{\mathcal{A}}(m-1, n))$.

Шаг 3 выполним согласно алгоритму из [6]. Приведем его краткое описание.

Для каждой машины $i \in \mathcal{M}$ построим независимое расписание S_i , начинающееся в момент 0 и такое, что операции работ $j \in \mathcal{N}$ выполняются подряд без простоев в одном и том же порядке, задаваемом перестановкой π . Из (47) следует, что каждая машина работает непрерывно в интервале $[0, M]$. В последующем эти расписания претерпят ряд сдвигов вправо, при этом часть расписания, выходящая за границу интервала $[0, M]$, будет отсекается и перемещаться в начало этого интервала. Таким образом, расписания работы машин $i \in \mathcal{M}$ удобно представлять находящимися на «барабане» с длиной окружности, равной M , где каждое расписание S_i представлено независимо вращающимся «кольцом» \mathcal{X}_i . При этом считаем, что кольцо \mathcal{X}_1 остается неподвижным и начало расписания S_1 задает точку отсчета времени.

Пусть k^* — (первое) значение k , на котором достигается максимум величины, стоящей в правой части (40) при $i = 1$. Таким образом,

$$\Delta_1 = \sum_{j=1}^{k^*} t_{\pi,1} - \sum_{j=1}^{k^*-1} t_{\pi,2}.$$

Выполним $m - 1$ последовательных поворотов (будем нумеровать их от 2 до m) колец $\mathcal{X}_2, \dots, \mathcal{X}_m$, оставляя кольцо \mathcal{X}_1 на месте.

Поворот 2. Повернем каждое кольцо $\mathcal{X}_2, \dots, \mathcal{X}_m$ вправо на величину Δ_1 , сохраняя неизменным их взаимное расположение. При этом момент $t^* \doteq \sum_{j=1}^{k^*} t_{\pi,1}$ совпадет с моментами окончания операции работы π_{k^*} на машине 1 и операции работы π_{k^*-1} на машине 2. Момент t^* принимаем за будущую точку разрезания расписания на «барабане».

Поворот ν ($\nu = 3, \dots, m$). Повернем каждое кольцо $\mathcal{X}_\nu, \dots, \mathcal{X}_m$ вправо на величину $\Delta_{\nu-1}$ и затем — еще на минимальную величину $\delta_{\nu-1} \geq 0$ (дополнительный поворот), достаточную для того, чтобы момент t^* совпал с моментом окончания какой-то операции машины ν . Ясно, что

$$\delta_\nu < K, \nu = 2, \dots, m - 1. \quad (50)$$

Суммарный (по всем $m - 1$ поворотам) поворот кольца \mathcal{X}_m вправо относительно кольца \mathcal{X}_1 равен $\Delta' \doteq \sum_{i=1}^{m-1} \Delta_i + \sum_{i=2}^{m-1} \delta_i$, что эквивалентно его повороту влево на $M - \Delta'$ или повороту кольца \mathcal{X}_1 вправо относительно \mathcal{X}_m на эту же величину. Нетрудно убедиться, что если

$$M - \Delta' \geq \Delta_m, \quad (51)$$

то полученное в результате всех поворотов колец расписание S' будет допустимым на барабане. Это означает, что интервалы выполнения любых двух операций одной и той же работы не пересекаются.

Теперь для получения из S' допустимого расписания длины M достаточно разрезать все кольца в точке t^* . Поскольку при этом ни для какой операции интервал ее выполнения не будет разрезан (что обеспечено величинами $\{\delta_\nu\}$ дополнительных поворотов), полученное расписание является допустимым. Очевидно, что оно будет допустимым и для исходной матрицы (более коротких) длительностей операций, имеющейся на входе до шага 1.

Алгоритм \mathcal{A}_1 описан.

Из (49)–(51) следует, что для построения указанным выше способом допустимого расписания длины M достаточно выполнения условия (39). При этом алгоритм \mathcal{A}_1 имеет декларируемую в теореме оценку трудоемкости. Теорема 2 доказана.

Из теоремы 2 и следствия 1 теоремы 1 вытекает

Теорема 3. Существует алгоритм трудоемкости $O(n \log n)$, который для всякого входа задачи OS с тремя машинами и n работами, удовлетворяющего условию

$$M \geq 7K, \quad (52)$$

строит оптимальное расписание длины M .

Заметим, что построение оптимального расписания задачи OS(3) алгоритмом \mathcal{A}_1 возможно и в тех случаях, когда вход задачи не удовлетворяет условию (52), но удовлетворяет (51). Во всех прочих случаях, как показано в [4], применение вместо алгоритма \mathcal{A}_1 простого жадного алгоритма позволяет с линейной от n трудоемкостью построить приближенное расписание S длины $T(S) \leq M + 2K$.

Из теоремы 3 и соотношения (4) вытекает

Следствие 4. Для минимальной функции $\eta^*(m)$, гарантирующей СБОР-свойство в задаче OS, при $m = 3$ справедливы оценки

$$4 \leq \eta^*(3) \leq 7.$$

§ 4. Три задачи типа job-shop

В этом параграфе рассмотрены задачи FS, AF(3) и R231. В случае произвольного числа машин m задача FS сводится к задаче HCB2($m-1$), что с учетом леммы 2 обеспечивает ее сводимость к задаче HCB1($m-2$). Далее будет показано сведение задач AF(3) и R231 к задачам HCB3 и HCB4 соответственно. С учетом следствий 1–3 из теоремы 1 это позволяет сконструировать для трех рассматриваемых задач теории расписаний эффективные алгоритмы построения приближенных расписаний с рекордными оценками точности, а также уточнить границы минимальных интервалов локализации оптимумов этих задач.

При решении каждой задачи будем предполагать выполненными соотношения $K = 1$ (что достигается выбором K за новую единицу измерения времени) и (47) (что достигается выполнением шага 1 алгоритма \mathcal{A}_1 из § 3.) Таким образом, семейство векторов $\{d_1, \dots, d_n\} = D$, определенное согласно (41), становится \hat{s} -семейством векторов в пространстве R^{m-1} .

Задача flow-shop

Теорема 4. Пусть имеется алгоритм \mathcal{A}' , который при любых натуральных n и m для любого \hat{s} -семейства $\{x_1, \dots, x_n\} \subset R^m$ с трудоемкостью $\mathcal{T}_{\mathcal{A}'}(m, n)$ решает задачу HCB2(m) с целевой функцией

$$\theta_2(b_m) \leq \theta_2^*(m).$$

Тогда существует алгоритм \mathcal{A}_2 , который по всякому входу задачи FS с m машинами и n работами с трудоемкостью $O(\mathcal{T}_{\mathcal{A}'}(m-1, n) + mn)$ находит перестановочное расписание S_π длины

$$T(S_\pi) \leq M + (m-1 + \theta_2^*(m-1))K.$$

ДОКАЗАТЕЛЬСТВО. В [1] установлена следующая оценка длины любого перестановочного расписания S_π :

$$T(S_\pi) \leq M + \sum_{i=1}^{m-1} \max_{k=1, \dots, n} \left(\sum_{j=1}^k t_{\pi_j, i} - \sum_{j=1}^{k-1} t_{\pi_j, i+1} \right).$$

В терминах координат векторов $\{d_1, \dots, d_n\} = D$ и с использованием (40), (44), (45) эта оценка записывается в виде

$$\begin{aligned} T(S_\pi) &\leq M + \sum_{i=1}^{m-1} \Delta_i \leq M + (m-1)K \\ &+ \sum_{i=1}^{m-2} \max_{k=1, \dots, n} \min \{d_\pi^{k-1}(i) - d_\pi^{k-1}(i+1), d_\pi^k(i) - d_\pi^k(i+1)\} \\ &+ \max_{k=1, \dots, n} \min \{d_\pi^{k-1}(m-1), d_\pi^k(m-1)\}. \end{aligned} \quad (53)$$

Применив к семейству $D \subset R^{m-1}$ алгоритм \mathcal{A}' решения задачи НСВ2, найдем перестановку $\pi = (\pi_1, \dots, \pi_n)$, задающую нестрогое суммирование векторов из D в семействе полупространств $\mathcal{P}_2(m-1, b_{m-1}K)$, где $b_{m-1} = (\beta_1, \dots, \beta_{m-1})$,

$$\sum_{i=1}^{m-1} \beta_i \leq \theta_2^*(m-1). \quad (54)$$

Из (53), (54) заключаем, что длина $T(S_\pi)$ расписания S_π , построенного по найденной перестановке π , удовлетворяет оценке

$$T(S_\pi) \leq M + (m-1)K + \sum_{i=1}^{m-1} \beta_i K \leq M + (m-1 + \theta_2^*(m-1))K.$$

Теорема 4 доказана.

Из теоремы 4 и леммы 2 непосредственно вытекает

Теорема 5. Пусть имеется алгоритм \mathcal{A} , который при любых натуральных n и m для любого \hat{s} -семейства $\{x_1, \dots, x_n\} \subset R^m$ с трудоемкостью $\mathcal{T}_{\mathcal{A}}(m, n)$ решает задачу НСВ1(m) с целевой функцией

$$\theta_1(b_{m+1}) \leq \theta_1^*(m).$$

Тогда существует алгоритм \mathcal{A}'_2 , который по всякому входу задачи FS с m машинами и n работами с трудоемкостью $O(\mathcal{T}_\mathcal{A}(m-2, n) + mn)$ находит перестановочное расписание S_π длины

$$T(S_\pi) \leq M + (m - 1 + \theta_1^*(m - 2))K.$$

С учетом последней теоремы и следствия 1 теоремы 1 получаем следующие результаты.

Теорема 6. Существует алгоритм трудоемкости $O(n \log n)$, который по всякому входу задачи flow-shop с четырьмя машинами и n работами строит перестановочное расписание S_π длины

$$T(S_\pi) \leq M + 6K.$$

Следствие 5. Для любого входа задачи FS(m) ее перестановочный оптимум находится в интервале $[M, M + \tilde{\mu}(m)K]$, где значение функции $\tilde{\mu}(m)$ при $m = 4$ удовлетворяет оценкам

$$4 \leq \tilde{\mu}(4) \leq 6.$$

В случае трех машин задача FS, как явствует из теоремы 5, сводится к решению задачи НСВ1 в одномерном пространстве, т. е. к нахождению нестрогого суммирования \hat{s} -семейства чисел (не превосходящих 1 и с нулевой суммой) в семействе одномерных полупространств $\{P(-e_2, \beta_2), P(e_2, \beta_1)\}$ с минимальным значением $\beta_1 + \beta_2$. Иными словами, требуется нестрогое просуммировать числа внутри отрезка минимальной длины.

Нетрудно понять, что для решения этой задачи можно предложить приближенный алгоритм линейной трудоемкости, гарантирующий нахождение нестрогого суммирования чисел из любого заданного \hat{s} -семейства внутри любого отрезка длины 1, содержащего точку нуль. Таким образом, для задачи FS(3) получаем следующий результат.

Теорема 7. Существует алгоритм трудоемкости $O(n)$, который по всякому входу задачи flow-shop с тремя машинами и n работами строит перестановочное расписание S_π длины

$$T(S_\pi) \leq M + 3K.$$

Этот результат улучшает как результат из [7] (алгоритм трудоемкости $O(n \log n)$, позволяющий находить расписание длины не более $M + 3K$), так и результат из [9] (алгоритм трудоемкости $O(n)$, позволяющий находить расписание длины не более $M + 4K$).

Задача Акерса — Фридмана для трех машин

Теорема 8. Пусть имеется алгоритм \mathcal{A} , который при любом натуральном n для любого \hat{s} -семейства $\{x_1, \dots, x_n\} \subset R^2$ с трудоемкостью $\mathcal{T}_{\mathcal{A}}(n)$ решает задачу НСВЗ с гарантированной оценкой

$$\theta_3(b_6) \leq \theta_3^*.$$

Тогда существует алгоритм \mathcal{A}_3 , который по всякому входу задачи АФ с тремя машинами и n работами с трудоемкостью $O(\mathcal{T}_{\mathcal{A}}(n) + n \log n)$ находит расписание S длины

$$T(S) \leq M + (3 + \theta_3^*)K.$$

Доказательство. Алгоритм \mathcal{A}_3 состоит из двух шагов.

Алгоритм \mathcal{A}_3

Шаг 1. Применив к семейству векторов $D \subset R^2$, определенных согласно (41), алгоритм \mathcal{A} решения задачи НСВЗ, получим перестановку $\pi = (\pi_1, \dots, \pi_n)$, задающую нестрогое суммирование векторов D в семействе областей $\mathcal{P}_3(2, b_6)$ с оценкой $\theta_3(b_6) \leq \theta_3^*$. Для удобства выкладок далее считаем, что работы занумерованы согласно перестановке π , иначе говоря, считаем, что $\pi = (1, 2, \dots, n)$.

Шаг 2. Пусть σ_i^j обозначает i -ю операцию работы j . Определим приоритет \succ_p на множестве операций:

1) $\sigma_1^j \succ_p \sigma_3^k, \sigma_2^j \succ_p \sigma_3^k$ при всех $j, k \in \mathcal{N}$, т. е. любая первая или вторая операция какой-либо работы приоритетнее любой третьей;

2) $\sigma_i^j \succ_p \sigma_\nu^k$ при $\nu \in \{1, 2\}$ и всех j, k таких, что $j < k$, т. е. на множестве первых и вторых операций приоритет определяется перестановкой π номеров работ;

3) $\sigma_3^j \succ_p \sigma_3^k$ при всех j, k таких, что $j < k$.

Расписание S построим жадным алгоритмом, который

- не позволяет машине простаивать, если есть операции, готовые на ней выполняться;
- выбирает очередную операцию на освободившуюся машину из очереди ждущих операций согласно определенному выше приоритетному порядку \succ_p .

Алгоритм \mathcal{A}_3 описан.

Для каждой машины $i = 1, 2, 3$ очередь ждущих операций организуем в виде «кучи» [10], т. е. такого двоичного дерева T^i с вершинами-операциями, в котором приоритет операций убывает по каждой цепочке, идущей из корня в листья дерева T^i , и длина каждой такой цепочки не превосходит $O(\log n)$. Тогда постановка каждой операции в очередь

ждуших и удаление наиболее приоритетной операции из очереди выполняются за $O(\log n)$ элементарных действий. Это обеспечивает реализацию жадного алгоритма с трудоемкостью $O(n \log n)$ и тем самым позволяет получить требуемую в теореме оценку трудоемкости алгоритма. Убедимся в справедливости верхней оценки для длины расписания.

Расписание работы произвольной машины i можно представить как чередование интервалов работы $[s_\nu^i, f_\nu^i)$ и интервалов простоя $[f_\nu^i, s_{\nu+1}^i)$, $\nu = 1, 2, \dots$

Пусть $o_{ji} = o_1^j$ — первая операция работы j . Так как она стоит в очереди на машину i с момента $t = 0$, то она не может пропустить вперед какую-либо менее приоритетную операцию. Поэтому для момента f_{ji} ее окончания мы можем выписать оценку

$$f_{ji} \leq \sum_{q=1}^j t_{qi} \doteq T_i^j. \quad (55)$$

Пусть $o_{ji} = o_2^j$ — вторая операция работы j , и она выполняется на машине i в интервале $[s_{ji}, f_{ji}) \subseteq [s_\nu^i, f_\nu^i)$. Пусть $o_{j_1i}, o_{j_2i}, \dots, o_{j_\nu i} = o_{ji}$ — максимальная цепочка операций, последовательно выполняемых на машине i в интервале $[s_\nu^i, f_{ji})$, такая, что $o_{j_1i} \succ_p o_{j_2i} \succ_p \dots \succ_p o_{ji}$. Операция o_{j_1i} либо является первой операцией в интервале $[s_\nu^i, f_{ji})$, т. е. $s_{j_1i} = s_\nu^i$, либо ей предшествует менее приоритетная операция. В обоих случаях между моментом постановки o_{j_1i} в очередь на машину i и моментом s_{j_1i} прошло менее $K = 1$ единиц времени.

Из определения приоритета \succ_p заключаем, что $o_{j_1i} \in \{o_1^{j_1}, o_2^{j_1}\}$ и

$$j_1 < j_2 < \dots < j_\nu = j. \quad (56)$$

Пусть $o_{j_1i} = o_2^{j_1}$, т. е. o_{j_1i} — вторая операция работы j_1 , первая операция которой выполняется на машине $i_1 \neq i$. Тогда из (55) следует, что

$$s_{j_1i} < f_{j_1i_1} + 1 \leq T_{i_1}^{j_1} + 1 = T_i^{j_1-1} + 1 + (T_{i_1}^{j_1} - T_i^{j_1-1}).$$

Отсюда с учетом (56) получаем верхнюю оценку момента окончания операции o_{ji} :

$$f_{ji} = s_{j_1i} + \sum_{k=1}^{\nu} t_{j_k i} < T_i^j + 1 + \max_k (T_{i_1}^k - T_i^{k-1}). \quad (57)$$

В этой оценке мы отошли от ее происхождения от работы j_1 , перейдя к $\max_{k \in \mathcal{N}}$. Поэтому и о машине i_1 можно сказать лишь то, что $i_1 \neq i$.

Поскольку величина \max_k в правой части (57) неотрицательна, то оценка (57) справедлива и в случае $o_{j_1i} = o_1^{j_1}$. Таким образом, правая

часть из (57) является верхней оценкой момента окончания второй операции любой работы. Следовательно, момент \hat{t}_{ji_3} постановки работы j в очередь на машину i_3 (для выполнения ее третьей операции) удовлетворяет соотношениям

$$\begin{aligned}\hat{t}_{ji_3} &= f_{ji_2} < T_{i_2}^j + 1 + \max_k (T_{i_1}^k - T_{i_2}^{k-1}) \\ &= T_{i_3}^{j-1} + 1 + (T_{i_2}^j - T_{i_3}^{j-1}) + \max_k (T_{i_1}^k - T_{i_2}^{k-1}) \leq T_{i_3}^{j-1} + 1 + \beta,\end{aligned}\quad (58)$$

где

$$\beta = \max_{\{i', i'', i''', i'''\} | i'' \neq i', i'' \neq i'''} \left\{ \max_k (T_{i''}^k - T_{i'''}^{k-1}) + \max_k (T_{i'}^k - T_{i''}^{k-1}) \right\}.$$

Сравнивая (58) с (57), нетрудно убедиться, что момент постановки второй операции любой работы в очередь на свою машину также удовлетворяет оценке (58), не говоря уже о первых операциях, для которых $\hat{t}_{ji} = 0$. Таким образом, момент \hat{t}_{ji} постановки в очередь на машину i любой операции o_{ji} удовлетворяет неравенству

$$\hat{t}_{ji} < T_i^{j-1} + 1 + \beta. \quad (59)$$

Пусть i — произвольная машина. Покажем, что длительность D_i ее простоя не превосходит $1 + \beta$. Если момент t' начала последнего интервала работы машины i наступает раньше $1 + \beta$, то, очевидно, $D_i < 1 + \beta$. Пусть $t' \geq 1 + \beta$ и j^* — максимальное j такое, что

$$T_i^{j^*-1} + 1 + \beta \leq t'. \quad (60)$$

Из (59) и (60) заключаем, что все операции o_{1i}, \dots, o_{j^*i} встали в очередь на машину i раньше момента t' . Так как моменту t' непосредственно предшествует простой на машине i , то все операции, вставшие в очередь на машину i строго до момента t' , успели и выполняться до момента t' согласно первому правилу жадного алгоритма. Отсюда $j^* < n$, и длительность D_i простоя машины i до момента t' (а значит, и до момента C_i окончания ее работы) удовлетворяет неравенствам

$$D_i \leq t' - T_i^{j^*} < 1 + \beta \quad (61)$$

по определению j^* .

Оценим величину β через величины β_i , определенные в задаче НСВЗ. Для любых $j \in \mathcal{N}$ и $\nu', \nu'' \in \mathcal{M}$, $\nu' \neq \nu''$, из определения векторов $\{a_i \mid i \in \mathcal{N}_6\}$ и $\{d_j \mid j \in \mathcal{N}\}$ имеем $t_{j\nu'} - t_{j\nu''} = (a_i, d_j)$, где соответствие между i, ν' и ν'' устанавливается следующей таблицей:

i	1	2	3	4	5	6	1
ν'	1	2	2	3	3	1	1
ν''	3	3	1	1	2	2	3

Из таблицы видно, что если $t_{ji'} - t_{ji''} = (a_{i_1}, d_j)$, $t_{ji''} - t_{ji'''} = (a_{i_2}, d_j)$, то i_1 и i_2 не могут совпадать или отличаться на 1 (более точно, $i_1 \neq i_2$, $i_1 \neq s(i_2)$, $i_2 \neq s(i_1)$), если иметь в виду циклический порядок на N_6 , определенный в § 2). В противном случае во второй и третьей строках таблицы в одном столбце или в соседних столбцах по диагонали окажутся два совпадающих элемента i'' , что невозможно. (Легко заметить, что каждая пара элементов (i', i'') во второй строке таблицы граничит по диагонали с парами (i'', i'') , (i''', i''') из третьей строки, где i', i'', i''' попарно различны.) Таким образом, обозначив $d^k = \sum_{j=1}^k d_j$, получим верхнюю оценку величины β :

$$\begin{aligned} \beta = & \max_{\{i', i'', i''' \mid i'' \neq i', i'' \neq i'''\}} \left(\max_{k \in \mathcal{N}} \min \left\{ \sum_{q=1}^{k-1} (t_{qi''} - t_{qi'''}) + t_{ki''}, \right. \right. \\ & \left. \sum_{q=1}^k (t_{qi''} - t_{qi'''}) + t_{ki''} \right\} + \max_{k \in \mathcal{N}} \min \left\{ \sum_{q=1}^{k-1} (t_{qi'} - t_{qi''}) + t_{ki'}, \right. \\ & \left. \sum_{q=1}^k (t_{qi'} - t_{qi''}) + t_{ki''} \right\} \Bigg) \leq \max_{(i_1, i_2) \in I} \left(\max_{k \in \mathcal{N}} \min \{ (a_{i_2}, d^{k-1}), (a_{i_2}, d^k) \} \right. \\ & \left. + \max_{k \in \mathcal{N}} \min \{ (a_{i_1}, d^{k-1}), (a_{i_1}, d^k) \} \right) + 2, \end{aligned}$$

где $I = \{(i_1, i_2) \mid i_1, i_2 \in N_6, i_1 \neq i_2, i_1 \neq s(i_2), i_2 \neq s(i_1)\}$. Поскольку перестановка $\pi = (1, 2, \dots, n)$ задает нестрогое суммирование векторов из $D = \{d_1, \dots, d_n\}$ в каждой области $\{P(a_1, \beta_1), \dots, P(a_6, \beta_6)\}$, получаем

$$\beta \leq 2 + \max_{(i_1, i_2) \in I} (\beta_{i_1} + \beta_{i_2}) = 2 + \theta_3(b_6) \leq 2 + \theta_3^*.$$

С учетом (61) получаем оценку

$$C_i = M + D_i \leq M + 1 + \beta \leq M + 3 + \theta_3^*,$$

что дает требуемую оценку длины расписания. Теорема 8 доказана.

Из теоремы 8 и следствия 2 теоремы 1 вытекают следующие результаты.

Теорема 9. Существует алгоритм трудоемкости $O(n \log n)$, который по всякому входу задачи Акерса — Фридмана с тремя машинами и n работами строит расписание S длины

$$T(S) < M + 5.5K.$$

Следствие 6. Для любого входа задачи $AF(m)$ ее оптимум находится в интервале $[M, M + \mu_{AF}(m)K]$, где значение функции $\mu_{AF}(m)$ при $m = 3$ удовлетворяет оценкам

$$3 \leq \mu_{AF}(3) \leq 5.5.$$

Задача R231

Теорема 10. Пусть имеется алгоритм \mathcal{A} , который при любом натуральном n для любого \hat{s} -семейства $\{x_1, \dots, x_n\} \subset R^2$ с трудоемкостью $\mathcal{T}_{\mathcal{A}}(n)$ решает задачу НСВ4 с гарантированной оценкой

$$\theta_4(b_3) \leq \theta_4^*.$$

Тогда существует алгоритм \mathcal{A}_4 , который по всякому входу задачи R231 с n работами с трудоемкостью $O(\mathcal{T}_{\mathcal{A}}(n) + n \log n)$ находит расписание S длины

$$T(S) \leq M + (3 + \theta_4^*)K.$$

Доказательство. Для построения искомого расписания S применим алгоритм \mathcal{A}_4 , который отличается от алгоритма \mathcal{A}_3 лишь тем, что на шаге 1 вместо задачи НСВ3 решается задача НСВ4.

Пусть в результате решения задачи НСВ4 найдена перестановка векторов $\pi = (\pi_1, \dots, \pi_n)$, задающая нестрогое суммирование векторов из $D = \{d_1, \dots, d_n\} \subset R^2$ в семействе полуплоскостей $\mathcal{P}_4(2, b_3) = \{P(e_1 - e_2, \beta_1), P(e_2, \beta_2), P(-e_1, \beta_3)\}$, где

$$\theta_4(b_3) = \max\{\beta_1 + \beta_2, \beta_2 + \beta_3\} \leq \theta_4^*. \quad (62)$$

Вновь для удобства выкладок считаем, что $K = 1$ и работы перенумерованы согласно перестановке π , иначе говоря, что $\pi = (1, 2, \dots, n)$.

Аналогично тому, как это делалось для задачи AF(3), оценим сверху момент прихода каждой работы $j \in \mathcal{N}$ на место выполнения своей второй и своей третьей операций. Получаем, что

- всякая работа $j \in \mathcal{N}_1$ приходит на машину 3 раньше момента $T_3^{j-1} + 3 + \beta_1 + \beta_2$;
- всякая работа $j \in \mathcal{N}_2$ приходит на машину 3 не позже момента $T_3^{j-1} + 1 + \beta_2$ и на машину 1 — раньше момента $T_1^{j-1} + 3 + \beta_2 + \beta_3$.

Следовательно, машина 3 заканчивает работу не позднее момента $M + 3 + \beta_1 + \beta_2$, а машина 1 — не позднее момента $M + 3 + \beta_2 + \beta_3$, и с учетом (62) получаем требуемую оценку длины расписания S . Теорема 10 доказана.

Из теоремы 10 и следствия 3 теоремы 1 вытекают следующие результаты.

Теорема 11. Существует алгоритм трудоемкости $O(n \log n)$, который по всякому входу задачи R231 с n работами строит расписание S длины

$$T(S) < M + 5K.$$

Следствие 7. Для любого входа задачи R231 ее оптимум находится в интервале $[M, M + \mu_{R231} K]$, где константа μ_{R231} удовлетворяет оценкам

$$3 \leq \mu_{R231} \leq 5.$$

Автор благодарит Э. Х. Гимади и Н. И. Глебова за полезное обсуждение работы.

ЛИТЕРАТУРА

1. **Севастьянов С. В.** Нестрогое суммирование векторов в задачах теории расписаний // Сиб. журн. исслед. операций. 1994. Т. 1, № 2. С. 67–99.
2. **Fiala T.** An algorithm for the open-shop problem // Math. Oper. Res. 1983. V. 8, N 1. P. 100–109.
3. **Gonzalez T., Sahni S.** Open shop scheduling to minimize finish time // J. Assoc. Comput. Mach. 1976. V. 23, N 4. P. 665–679.
4. **Bárány I., Fiala T.** Nearly optimum solution of multimachine scheduling problems // Szigma — Mat.-Közgaz-dasági Folyóirat. 1982. V. 15, N 3. P. 177–191.
5. **Севастьянов С. В.** Полиномиально разрешимый случай задачи open-shop с произвольным числом машин // Кибернетика и системный анализ. 1992. № 6. С. 135–154.
6. **Sevast'janov S. V.** Vector summation in Banach space and polynomial algorithms for flow shops and open shops // Math. Oper. Res. 1995. V. 20, N 1. P. 90–103.
7. **Севастьянов С. В.** Алгоритмы с оценками для задач Джонсона и Акерса — Фридмана в случае трех станков // Управляемые системы. Новосибирск, 1982. Вып. 22. С. 51–57.
8. **Tarjan R. E.** Data structures and network algorithms. Philadelphia, PA: SIAM, 1983.
9. **Душин Б.И.** Замечание к алгоритму в одномаршрутной задаче Джонсона // Кибернетика. 1980. № 2. С. 129–131.
10. **Williams J. W. J.** Heapsort // Comm. ACM. 1964. V. 7, N 6. P. 347–348.

Адрес автора:

Россия,
630090 Новосибирск,
Университетский пр., 4,
Институт математики СО РАН

Статья поступила

20 февраля 1995 г.