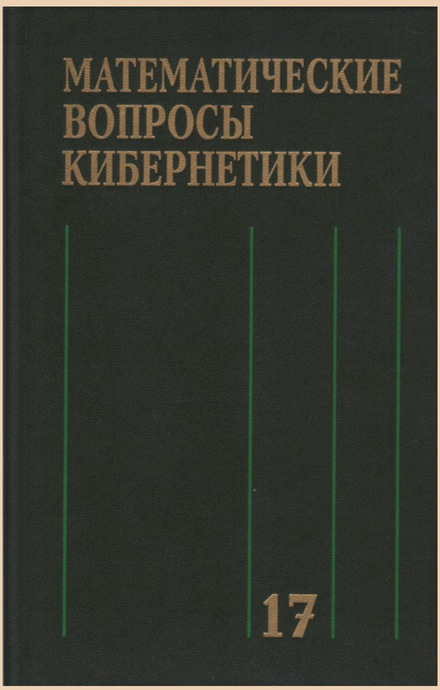

А. М. Зубков,
Д. В. Соколов

Алгоритмы частичной сортировки множеств сумм

Рекомендуемая форма библиографической ссылки: Зубков А. М., Соколов Д. В. Алгоритмы частичной сортировки множеств сумм // Математические вопросы кибернетики. Вып. 17. — М.: ФИЗМАТЛИТ, 2008. — С. 225–234. URL: http://library.keldysh.ru/mvk.asp?id=2008-225

АЛГОРИТМЫ ЧАСТИЧНОЙ СОРТИРОВКИ МНОЖЕСТВ СУММ

А. М. ЗУБКОВ *), **_Д. В. СОКОЛОВ_**

(МОСКВА)

Задача сортировки (упорядочения) элементов конечного множества хорошо известна; существует много различных алгоритмов ее решения. В работе предлагаются алгоритмы, позволяющие строить совокупности наибольших (или наименьших) элементов множества значений линейной функции на вершинах n-мерного куба или множества комбинаторных сумм по перестановкам. Алгоритмы используют аналитическую структуру сортируемых множеств сумм и не требуют создания массива, содержащего значения всех сумм.

Под алгоритмом частичной сортировки в настоящей работе понимается способ, позволяющий выписывать элементы числового множества в порядке неубывания. Разумеется, для решения задачи частичной сортировки достаточно отсортировать все числовое множество, однако такой способ может оказаться практически нереализуемым, если мощность этого множества слишком велика.

Общая простая идея алгоритмов частичной сортировки множества чисел F состоит в том, чтобы сначала найти максимальный элемент F и образовать из него множество Φ_1, состоящее из одного максимального элемента, а затем последовательно добавлять к нему по одному элементу, получая множества $\Phi_2, \Phi_3, \ldots$, состоящие из 2,3,... максимальных элементов множества F. Чтобы упростить переходы от Φ_k к Φ_{k+1}, можно параллельно с множествами Φ_k строить такие множества D_k, что D_k заведомо содержит максимальный элемент множества $F\backslash\Phi_k$. Тогда построение множества Φ_{k+1} сведется к поиску максимального элемента множества D_k. Алгоритм тем эффективнее, чем меньше элементов содержат множества D_k. Разумеется, структура множеств D_k сильно зависит от структуры множества F. Эта зависимость иллюстрируется примерами, рассмотренными в настоящей работе.

Ниже предлагаются алгоритмы частичной сортировки числовых множеств следующих двух видов:

А) множество $C = \left\{ \sum\limits_{j=1}^{n} \varepsilon_j c_j : \varepsilon_1, \ldots, \varepsilon_n \in \{0, 1\} \right\}$ всевозможных сумм заданных чисел $c_1, \ldots, c_n$,

*) Работа поддерживалась Программой РАН «Современные проблемы теоретической математики» и грантом НШ-4129.2006.1 Программы поддержки ведущих научных школ России.

Б) множество $D = \left\{\sum\limits_{j=1}^{n} a_j b_{\sigma_j} : \sigma = (\sigma_1, \ldots, \sigma_n) \in S_n\right\}$ всевозможных комбинаторных сумм, построенных по двум наборам чисел $a_1, \ldots, a_n$ и $b_1, \ldots, b_n$.

Задачи частичной сортировки множеств такого типа возникают при построении статистических критериев и доверительных множеств в задачах математической статистики. Примером множеств типа А являются логарифмы вероятностей результатов n независимых испытаний Бернулли с вероятностями успеха $p_1, \ldots, p_n$. В этом случае логарифм вероятности $P(e_1, \ldots, e_n)$ того, что результаты n испытаний описываются двоичным вектором $(e_1, \ldots, e_n)$, есть

$$\log \prod_{j=1}^{n} p_j^{e_1}(1-p_j)^{1-e_j} = e_1 \log \frac{p_1}{1-p_1} + \ldots + e_n \log \frac{p_n}{1-p_n} + \log \prod_{j=1}^{n}(1-p_j),$$

т. е. является линейной функцией от координат вектора $(e_1, \ldots, e_n)$. Множество исходов n испытаний, имеющее минимальную мощность и вероятность, не меньшую $1-\varepsilon$, должно состоять из наиболее вероятных исходов. Если совокупность результатов исходов не попадает в такое множество, то гипотезу о том, что вектор вероятностей исходов равен $\boldsymbol{p} = (p_1, \ldots, p_n)$, следует подвергнуть сомнению. Задача построения аналогичного множества в связи с решением систем уравнений с искаженными правыми частями рассматривалась в [1]. Метод, аналогичный описанному в настоящей работе, для множеств типа А предлагал ранее С. М. Пушич.

Суммы типа Б возникают в статистических критериях, основанных на перестановках наблюдений (см., например, [7]), или при использовании статистики Пирсона χ^2 для определения неизвестной перестановки σ исходов полиномиальной схемы с известными вероятностями $a_1, \ldots, a_n$ по частотам $b_{\sigma_1}, \ldots, b_{\sigma_n}$. В последнем случае для построения доверительного множества перестановок нужно найти все перестановки, для которых суммы вида Б принимают значения, не меньшие заданного, или найти подмножество, состоящее из заданного числа перестановок, с наибольшими значениями сумм.

Известно (см., например, [3, 4]), что сумма

$$\sum_{j=1}^{n} a_j b_{\sigma_j}$$

максимальна, если $a_1 \leqslant a_2 \leqslant \ldots \leqslant a_n$ и $b_{\sigma_1} \leqslant b_{\sigma_2} \leqslant \ldots \leqslant b_{\sigma_n}$, и минимальна, если $a_1 \leqslant a_2 \leqslant \ldots \leqslant a_n$ и $b_{\sigma_1} \geqslant b_{\sigma_2} \geqslant \ldots \geqslant b_{\sigma_n}$. В [5–7] доказана асимптотическая нормальность сумм типа Б при $n \to \infty$ в схемах серий для случайной подстановки σ, имеющей равномерное распределение на множестве всех подстановок порядка n.

Рассмотрим сначала суммы типа А.

Итак, пусть

$$V^n = \{0, 1\}^n = \{\boldsymbol{v} = (v_1, \ldots, v_n) : v_i \in \{0, 1\}, i = 1, \ldots, n\}$$

— множество вершин n-мерного единичного куба и

$$w(\boldsymbol{v}) = c_1 v_1 + \ldots + c_n v_n \tag{1}$$

— заданная на V^n линейная функция; без ограничения общности можно считать, что все c_i отрицательны, так что максимальное значение функция $w(\boldsymbol{v})$ принимает в точке $\boldsymbol{v} = (0, \ldots, 0)$.

Если задано число m и требуется построить множество

$$B(m) = \{\boldsymbol{v} \in V^n : w(\boldsymbol{v}) \geqslant m\},$$

то можно воспользоваться методом ветвей и границ. Каждой вершине $\boldsymbol{v}$, $\boldsymbol{v} = (v_1, \ldots, v_n) \in V^n$, взаимно однозначно сопоставим целое число (вес)

$$|\boldsymbol{v}| = 2^{n-1}v_n + \ldots + 2v_2 + v_1 \tag{2}$$

и вершины $\boldsymbol{y}_1(\boldsymbol{v}), \ldots, \boldsymbol{y}_n(\boldsymbol{v})$ с весами

$$|\boldsymbol{y}_k(\boldsymbol{v})| = \begin{cases} 2^{n-1}v_n + \ldots + 2^{k-1}v_k + 2^{k-1}, & \text{если } (v_n, \ldots, v_k) \neq (1, \ldots, 1), \\ 2^n - 1 & \text{в противном случае,} \end{cases}$$

$$k = 1, \ldots, n,$$

$$|\boldsymbol{y}_{n+1}(\boldsymbol{v})| = 2^n - 1.$$

Нетрудно проверить, что $|\boldsymbol{y}_k(\boldsymbol{v})| \leqslant |\boldsymbol{y}_{k+1}(\boldsymbol{v})|$ и что если $v_k = 1$ и $k < n$, то $\boldsymbol{y}_k(\boldsymbol{v}) = \boldsymbol{y}_{k+1}(\boldsymbol{v})$.

Вершины $\boldsymbol{y}$ из V^n с весами $|\boldsymbol{y}| \in (|\boldsymbol{y}_k(\boldsymbol{v})|, |\boldsymbol{y}_{k+1}(\boldsymbol{v})|)$ отличаются от вершины $\boldsymbol{y}_k(\boldsymbol{v})$ лишь координатами $v_1, \ldots, v_{k-1}$ (которые равны 0 для $\boldsymbol{y}_k(\boldsymbol{v})$). Поэтому для всех таких вершин $w(\boldsymbol{y}) < w(\boldsymbol{y}_k(\boldsymbol{v}))$, и если $w(\boldsymbol{y}_k(\boldsymbol{v})) < m$, то все эти вершины не принадлежат искомому множеству $B(m)$.

Сделанные замечания приводят к следующему алгоритму построения множества $B(m)$ методом ветвей и границ.

1. Если $w((0, \ldots, 0)) < m$, то $B(m) = \varnothing$. В противном случае полагаем $\boldsymbol{v}^{(1)} = (0, \ldots, 0)$, $B_1(m) = \{\boldsymbol{v}^{(1)}\}$ и $t = 1$, $b = 1$.

2. Если $\boldsymbol{v}^{(t)} \notin B(m) = \{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(b)}\}$, то работа закончена. В противном случае строим множество $A_t = \{k \in \{1, \ldots, n\} : w(\boldsymbol{y}_k(\boldsymbol{v}^{(t)})) \geqslant m\}$.

3. Элементы $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{|A_t|}$ множества A_t добавляем к множеству $B(m)$, полагая $\boldsymbol{v}^{(b+1)} = \boldsymbol{u}_1, \ldots, \boldsymbol{v}^{(b+|A_t|)} = \boldsymbol{u}_{|A_t|}$, и заменяем b на $b + |A_t|$ (при $A_t = \varnothing$ множество $B(m)$ и число b не изменяются). После этого увеличиваем t на 1 и возвращаемся к п. 2.

Этот алгоритм можно использовать для решения упоминавшейся во введении задачи о построении минимального множества исходов n испытаний Бернулли с суммарной вероятностью, не меньшей $1 - \varepsilon$, если удастся найти или достаточно точно оценить такую величину m, что

$$\sum_{(e_1,\ldots,e_n): P(e_1,\ldots,e_n) > m} P(e_1, \ldots, e_n) < 1 - \varepsilon \leqslant \sum_{(e_1,\ldots,e_n): P(e_1,\ldots,e_n) \geqslant m} P(e_1, \ldots, e_n);$$

оценки такого типа можно найти, например, в [2].

Теперь опишем алгоритм, позволяющий выписывать элементы $\boldsymbol{v}$ множества V^n в порядке невозрастания значений $w(\boldsymbol{v})$ и затрачивающий в среднем на построение каждого очередного элемента упорядоченного списка $O(n)$ операций. Очевидно, что такой алгоритм позволяет строить минимальное множество исходов n испытаний Бернулли с суммарной вероятностью, не меньшей $1-\varepsilon$, без оценок свойств искомого множества: достаточно в процессе построения последовательности исходов с монотонно невозрастающими вероятностями вычислять сумму вероятностей уже построенных исходов и остановиться в первый момент, когда эта сумма достигнет уровня $1-\varepsilon$.

Обозначим через

$$\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^{2^n} \tag{3}$$

упорядоченную по невозрастанию значений функции w последовательность элементов V^n:

$$w(\boldsymbol{v}^1) \geqslant w(\boldsymbol{v}^2) \geqslant \ldots \geqslant w(\boldsymbol{v}^{2^n});$$

элементы, на которых значения функции w одинаковы, будем упорядочивать по возрастанию их весов.

Установим свойства последовательности (3), которые используются при построении алгоритма. Обозначим через $\boldsymbol{e}_i$ элемент V^n, у которого все координаты, кроме i-й, равны 0, а i-я координата равна 1. Разобьем V^n на $n+1$ непересекающихся подмножеств

$$B_i = \{\boldsymbol{v} \in V^n : v_1 = \ldots = v_{n-i} = 0, v_{n-i+1} = 1\}, \quad i = 0, 1, \ldots, n,$$

и построим расширяющуюся последовательность множеств

$$C_i = \bigcup_{j=0}^{i} B_j = \{\boldsymbol{v} \in V^n : v_1 = \ldots = v_{n-i} = 0\}, \quad i = 0, \ldots, n.$$

Очевидно,

$$|B_0| = 1,\ |B_i| = 2^{i-1}\ (i = 1, \ldots, n), \quad |C_i| = 2^i\ (i = 0, \ldots, n),$$

и

$$\begin{aligned} &\boldsymbol{v} - \boldsymbol{e}_{n-i+1} \in C_{i-1}, \quad \text{если} \quad \boldsymbol{v} \in B_i \quad (i = 1, \ldots, n),\\ &\boldsymbol{v} + \boldsymbol{e}_{n-i+1} \in B_i, \quad \text{если} \quad \boldsymbol{v} \in C_{i-1} \quad (i = 1, \ldots, n). \end{aligned} \tag{4}$$

По последовательности (3) для каждого $i = 0, \ldots, n$ однозначно определяются монотонно возрастающие последовательности $b_i(t)$ $(t = 1, \ldots, |B_i|)$ и $c_i(t)$ $(t = 1, \ldots, |C_i|)$ мест, занимаемых в последовательности (3) элементами множеств B_i и C_i соответственно:

$$B_i = \{\boldsymbol{v}^{b_i(1)}, \ldots, \boldsymbol{v}^{b_i(|B_i|)}\}, \quad C_i = \{\boldsymbol{v}^{c_i(1)}, \ldots, \boldsymbol{v}^{c_i(|C_i|)}\},$$

так что

$$w(\boldsymbol{v}^{b_i(1)}) \geqslant \ldots \geqslant w(\boldsymbol{v}^{b_i(|B_i|)}), \quad w(\boldsymbol{v}^{c_i(1)}) \geqslant \ldots \geqslant w(\boldsymbol{v}^{c_i(|C_i|)}). \tag{5}$$

Таким образом, последовательность (3) разбивается на непересекающиеся подпоследовательности

$$\{\boldsymbol{v}^{b_0(1)}\}, \quad \{\boldsymbol{v}^{b_1(1)}\}, \quad \{\boldsymbol{v}^{b_2(1)}, \boldsymbol{v}^{b_2(2)}\}, \ldots, \{\boldsymbol{v}^{b_n(1)}, \ldots, \boldsymbol{v}^{b_n(2^{n-1})}\}, \tag{6}$$

а подпоследовательности

$$\{\boldsymbol{v}^{c_0(1)}\}, \quad \{\boldsymbol{v}^{c_1(1)}, \boldsymbol{v}^{c_1(2)}\}, \ldots, \{\boldsymbol{v}^{c_n(1)}, \ldots, \boldsymbol{v}^{c_n(2^n)}\}, \tag{7}$$

вкладываются одна в другую. Для каждой из этих подпоследовательностей функция w монотонно не возрастает. Более содержательные связи между последовательностями (6) и (7) описаны в следующих леммах.

Лемма 1. *При любых $i \in \{1, \ldots, n\}$ и $t \in \{1, \ldots, |B_i|\}$ справедливы соотношения*

$$\boldsymbol{v}^{b_i(t)} = \boldsymbol{e}_{n-i+1} + \boldsymbol{v}^{c_{i-1}(t)}. \tag{8}$$

Иначе говоря, каждый элемент i-й $(i = 1, \ldots, n)$ подпоследовательности (6) получается из соответствующего элемента $(i-1)$-й последовательности (7) добавлением вектора $\boldsymbol{e}_{n-i+1}$.

Утверждение леммы следует из соотношений (4), (5) и из того, что

$$w(\boldsymbol{v}+\boldsymbol{e}_{n-i+1})=w(\boldsymbol{v})+w(\boldsymbol{e}_{n-i+1}) \quad \text{при} \quad \boldsymbol{v}\in C_{i-1}.$$

Лемма 2. *Пусть числа* $t\in\{1,\ldots,2^n-1\}, i\in\{1,\ldots,n\}, u\in\{1,\ldots,|B_i|\}$ *таковы, что*

$$\boldsymbol{v}^{t+1}=\boldsymbol{v}^{b_i(u)}. \tag{9}$$

а) *Если* $\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^t\}\cap B_i=\varnothing$, *то* $\boldsymbol{v}^{t+1}=\boldsymbol{e}_{n-i+1}$.

б) *Если* $\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^t\}\cap B_i\neq\varnothing$ *и* $k_i(t)=\max\{k:b_i(k)\leqslant t\}$, *а* $d_i(t)$ *определяется соотношением*

$$\boldsymbol{v}^{d_i(t)}=\boldsymbol{v}^{b_i(k_i(t))}-\boldsymbol{e}_{n-i+1}\in C_{i-1}, \tag{10}$$

то $u=k_i(t)+1$ *и*

$$\boldsymbol{v}^{t+1}=\boldsymbol{e}_{n-i+1}+\boldsymbol{v}^{f_i(t)},$$

где

$$f_i(t)=\min\{f\in\{d_i(t)+1,\ldots,t\}:\boldsymbol{v}^f\in C_{i-1}\}. \tag{11}$$

Доказательство. Утверждение а) очевидно, так как на элементе $\boldsymbol{e}_i$ функция w принимает наибольшее значение в множестве B_i, и поэтому первым элементом множества B_i в последовательности (3) может быть только $\boldsymbol{e}_i$.

Чтобы доказать утверждение б), заметим, что в силу условий леммы 2 элемент $\boldsymbol{v}^{t+1}$, $\boldsymbol{v}^{t+1}=\boldsymbol{v}^{b_i(u)}\in B_i$, в последовательности $\{\boldsymbol{v}^{b_i(1)},\ldots,\boldsymbol{v}^{b_i(|B_i|)}\}$ должен следовать за элементом $\boldsymbol{v}^{k_i(t)}$, т. е. $u=k_i(t)+1$. Далее, определенный равенством (10) элемент $\boldsymbol{v}^{d_i(t)}$ должен принадлежать множеству $\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^t\}$, так как в силу линейности функции w и ее невозрастания по каждой координате

$$w(\boldsymbol{v}^{d_i(t)})=w(\boldsymbol{v}^{b_i(k_i(t))}-\boldsymbol{e}_{n-i+1})=w(\boldsymbol{v}^{b_i(k_i(t))})-w(\boldsymbol{e}_{n-i+1})\geqslant w(\boldsymbol{v}^{b_i(k_i(t))}).$$

По тем же причинам элемент $\boldsymbol{v}^{t+1}-\boldsymbol{e}_{n-i+1}$ равен $\boldsymbol{v}^{b_i(u)}-\boldsymbol{e}_{n-i+1}$, принадлежит множеству $\{\boldsymbol{v}^1,\ldots,\boldsymbol{v}^t\}\cap C_{i-1}$ и в последовательности $\{\boldsymbol{v}^{c_{i-1}(1)},\ldots,\boldsymbol{v}^{c_{i-1}(|C_{i-1}|)}\}$ следует за элементом $\boldsymbol{v}^{d_i(t)}$. Значит, число $f_i(t)$ определено корректно. Из этих замечаний и из леммы 1 следует утверждение б). Лемма доказана.

Леммы 1 и 2 позволяют по любому начальному отрезку $v^1,\ldots,v^t$ последовательности (3) для каждого i, $i\in\{1,\ldots,n\}$, либо построить элемент $e_{n-i+1}+v^f$ из $B_i(t)=B_i\cap\{v^1,\ldots,v^t\}$ с максимальным в $B_i(t)$ значением функции w, либо убедиться в том, что $v^{t+1}\notin B_i$ (если не существует числа $f_i(t)$, удовлетворяющего условию (11)). Тогда v^{t+1} совпадает с тем из элементов вида $e_{n-i+1}+v^f$, построенных по начальному отрезку $v^1,\ldots,v^t$ последовательности (3), на котором функция w максимальна (а при равенстве значений w на нескольких элементах — с вектором, имеющим минимальный вес).

Заметим еще, что определенные формулами (10) и (11) последовательности $d_i(t)$ и $f_i(t)$ монотонно возрастают. Поэтому для нахождения очередных элементов этих последовательностей можно не просматривать каждый раз весь отрезок $v^1,\ldots,v^t$, а сдвигаться по нему в одном и том же направлении.

Опишем алгоритм построения последовательности (3).

1. Положить $\boldsymbol{v}^1=(0,\ldots,0)\in V^n$, $d_i=f_i=0$ $(i=1,\ldots,n)$, $t=1$.

2. Для каждого $i=1,\ldots,n$ положить

$$\boldsymbol{v}^{t+1}(i)=\begin{cases}\boldsymbol{e}_{n-i+1}+\boldsymbol{v}^{d_i}, & \text{если } d_i>0, f_i>0,\\ (1,\ldots,1), & \text{если } d_i>0, f_i=0,\\ \boldsymbol{e}_{n-i+1}, & \text{если } d_i=0.\end{cases}$$

3. Вычислить $w_i=w(\boldsymbol{v}^{t+1}(i))$, $i=1,\ldots,n$, и найти максимальное $i=i(t)$, на котором достигается $\max\{w_1,\ldots,w_n\}$.

4. Положить $\boldsymbol{v}^{t+1}=\boldsymbol{v}^{t+1}(i(t))$.

5. Если $\boldsymbol{v}^{t+1}(i(t))=\boldsymbol{e}_{i(t)}$, то положить $d_{i(t)}=1$. Если $d_{i(t)}>0$ и $f_{i(t)}>0$, то положить $d_{i(t)}=f_{i(t)}$.

6. Если $D_{i(t)}=\{d\in\{d_{i(t)}+1,\ldots,t+1\}:\boldsymbol{v}^d\in C_{i(t)-1}\}=\varnothing$, то положить $f_{i(t)}=0$, в противном случае положить $f_{i(t)}=\min\{d:d\in D_{i(t)}\}$.

Для каждого такого $i>i(t)$, что $f_i=0$, положить $f_i=t+1$.

7. Положить $t=t+1$. Если $t\leqslant 2^n$ и искомая часть последовательности (3) еще не построена, то вернуться к п. 2, в противном случае закончить работу.

Корректность описанного алгоритма следует из лемм 1 и 2. Чтобы оценить число операций, достаточно заметить, что для каждого t выполняется $O(n)$ операций, связанных с непосредственным построением $\boldsymbol{v}^{t+1}$, а общее число операций, выполненных к моменту построения $\boldsymbol{v}^t$ и связанных с вычислением d_i и f_i, имеет порядок $O(nt)$. Поэтому построение всей последовательности (3) проводится за $O(n2^n)$ операций; если же ограничиваться построением совокупности первых M элементов последовательности (3), то число операций не превзойдет $O(nM)$.

Этот алгоритм нетрудно обобщить на случай сортировки значений линейной функции, заданной на множестве вида

$$\{0,\ldots,m_1\}\times\{0,\ldots,m_2\}\times\ldots\times\{0,\ldots,m_n\},$$

если для каждого $i=1,\ldots,n$ рассматривать не одно множество B_i, а m_i множеств

$$B_{ir}=\{\boldsymbol{v}\in V^n: v_1=\ldots=v_{n-i}=0, v_{n-i+1}=r\},\quad r=1,\ldots,m_i.$$

Недостатком алгоритма (отличающим его от простого алгоритма, основанного на методе ветвей и границ) является необходимость хранить в памяти значительную часть построенной последовательности (3), начиная от минимального значения d_i.

Рассмотрим теперь метод частичной сортировки сумм типа Б.

Пусть даны два набора действительных чисел $\boldsymbol{a}=\{a_1,\ldots,a_n\}$ и $\boldsymbol{b}=\{b_1,\ldots,b_n\}$. С каждой перестановкой $\sigma=(\sigma_1,\ldots,\sigma_n)$, $\sigma\in S_n$, элементов множества $N_n=\{1,\ldots,n\}$ свяжем величину

$$c_{\boldsymbol{ab}}(\sigma)=\sum_{i=1}^n a_i b_{\sigma_i}. \tag{12}$$

Прямолинейный метод частичной сортировки множества сумм $c_{\boldsymbol{ab}}(\sigma)$, $\sigma\in S_n$, состоит в сортировке этого множества из $n!$ элементов, т. е. построении ряда

$$c_{\boldsymbol{ab}}(\nu^{(1)})\geqslant c_{\boldsymbol{ab}}(\nu^{(2)})\geqslant\ldots\geqslant c_{\boldsymbol{ab}}(\nu^{(n!)}); \tag{13}$$

число арифметических операций при использовании такого метода будет не меньше $n!n = O((n/e)^{n+2})$. Далее будет описан метод построения начальных отрезков вариационного ряда сумм $c_{\boldsymbol{ab}}(\sigma)$, который затрачивает $O(nT^2)$ арифметических операций на построение начального отрезка длиной T.

Опишем идею алгоритма. Для простоты будем предполагать, что все элементы множеств $a_1, \ldots, a_n$ и $b_1, \ldots, b_n$ попарно различны и

$$a_1 > \ldots > a_n, \quad b_1 > \ldots > b_n. \tag{14}$$

Если условия (14) не выполняются (т. е. среди чисел $a_1, \ldots, a_n$ или среди чисел $b_1, \ldots, b_n$ есть одинаковые), то можно обеспечить их выполнение за счет сколь угодно малого изменения значений некоторых из этих чисел; при этом значения сумм $c_{\boldsymbol{ab}}(\sigma)$ тоже изменятся сколь угодно мало.

Будем обозначать через $\varepsilon(A)$ перестановку элементов множества $A \subset \subset N_n$, $N_n = \{1, \ldots, n\}$, в которой элементы множества A стоят в порядке возрастания. Если множества $A = \{i_1, \ldots, i_s\}$, $B = \{j_1, \ldots, j_t\}$, $B \subset N_n$ не пересекаются, и $\zeta = (\zeta_1, \ldots, \zeta_s)$ — перестановка множества A, а $\tau = (\tau_1, \ldots \ldots, \tau_t)$ — перестановка множества B, то записью (ζ, τ) будем обозначать перестановку $(\zeta_1, \ldots, \zeta_s, \tau_1, \ldots, \tau_t)$ элементов множества $A \cup B$.

Расположим все перестановки элементов множества $N_n = \{1, \ldots, n\}$ в виде дерева Σ, считая его корнем перестановку $\varepsilon(N_n) = (1, \ldots, n)$; вершинам первого слоя сопоставим подстановки $(2, \varepsilon(N_n \backslash \{2\}))$ и $(\varepsilon(N_{i-1}), i + + 1, i, \varepsilon(N_n \backslash N_{i+1}))$, $i = 2, \ldots, n-1$. В общем случае для того, чтобы определить вершины дерева Σ, соединенные ребрами с вершиной $\sigma = (\sigma_1, \ldots, \sigma_n)$ предыдущего слоя, нужно найти минимальное i, для которого имеет место представление

$$\sigma = (\sigma_1, \ldots, \sigma_i, \varepsilon(\{\sigma_{i+1}, \ldots, \sigma_n\})),$$

и считать, что σ соединена ребрами с вершиной

$$(\sigma_1, \ldots, \sigma_{i-1}, \theta_i(\sigma), \varepsilon(\{\sigma_i, \ldots, \sigma_n\} \backslash \theta_i(\sigma))),$$

где $\theta_i(\sigma) = \min\{k > \sigma_i : k \notin \{\sigma_1, \ldots, \sigma_i\}\}$, и с вершинами

$$(\sigma_1, \ldots, \sigma_{j-1}, \eta_j(\sigma), \varepsilon(\{\sigma_j, \ldots, \sigma_n\} \backslash \eta_j(\sigma))), \quad j = i+1, \ldots, n-1,$$

где $\eta_j(\sigma) = \min\{s \in \{\sigma_{j+1}, \ldots, \sigma_n\} : s > \min\{\sigma_j, \ldots, \sigma_n\}\}$.

Хорошо известно (см., например, [2, стр.314–315] или лемму 3 ниже), что при условиях (14) максимальное значение $c_{\boldsymbol{ab}}(\nu)$ по всем $\nu \in S_n$ достигается на монотонной перестановке $\varepsilon(N_n) = (1, 2, \ldots, n)$. Из доказываемых ниже утверждений следует, что при условиях (14) перестановки $\nu^{(1)}, \ldots, \nu^{(t)}$, соответствующие любому начальному отрезку ряда (13), образуют поддерево Σ_t с корнем $\varnothing$ дерева Σ, получающееся из Σ отсечением некоторых (концевых) поддеревьев. Поэтому если $\nu^{(1)}, \ldots, \nu^{(t)}$ — начало ряда (13) и Σ_t — соответствующее ему поддерево, то $\nu^{(t+1)}$ принадлежит совокупности тех вершин дерева Σ, которые соединены ребрами с вершинами из Σ_t (т. е. с вершинами из границы множества Σ_t), и при переходе от $\nu^{(1)}, \ldots, \nu^{(t)}$ к $\nu^{(1)}, \ldots, \nu^{(t+1)}$ достаточно выбрать ту перестановку ν из вершин, граничащих с вершинами из множества Σ_t, для которой значение $c_{\boldsymbol{ab}}(\nu)$ максимально. Следовательно, при построении начального отрезка ряда (13) можно лишь строить границы множеств Σ_t при $t = 1, 2, \ldots$ и вычислять значения $c_{\boldsymbol{ab}}(\nu)$ для перестановок ν, входящих в эти границы.

Перейдем к более детальному изложению.

Л е м м а 3. *Если выполнены условия* (14), *то*

$$\max_{\nu\in S_n} c_{\boldsymbol{ab}}(\nu)=c_{\boldsymbol{ab}}(\varepsilon(N_n))=\sum_{i=1}^n a_ib_i,$$

$$\min_{\nu\in S_n} c_{\boldsymbol{ab}}(\nu)=c((n,n-1,\ldots,1))=\sum_{i=1}^n a_ib_{n-i+1}.$$

Д о к а з а т е л ь с т в о, например, первого утверждения основано на том, что от любой перестановки $\nu^{(0)}=(\nu_1,\ldots,\nu_n)$ можно перейти к перестановке $(1,2,\ldots,n)$ с помощью $n-1$ транспозиций или тождественных преобразований: k-е преобразование меняет местами k-й элемент перестановки с одним из последующих, так что после k-го преобразования k-й элемент становится равным k, а перестановка принимает вид $\nu^{(k)}=(1,\ldots,k-1,k,\nu_{k,k+1},\ldots,\nu_{k,n})$. Если преобразование с номером k не тождественное, то оно изменяет в сумме $c_{\boldsymbol{ab}}(\nu^{(k-1)})$ только два слагаемых: $a_kb_{\nu_{k-1,k}}+a_jb_k$ (число $j>k$ определяется условием $\nu_{k-1,j}=k$) заменяется на $a_kb_k+a_jb_{\nu_{k-1,k}}$. Так как при этом $a_k>a_j$ и $b_k>b_{\nu_{k-1,k}}$, то

$$a_kb_k+a_jb_{\nu_{k-1,k}}=a_kb_{\nu_{k-1,k}}+a_jb_k+(a_k-a_j)(b_k-b_{\nu_{k-1,k}})>a_kb_{\nu_{k-1,k}}+a_jb_k,$$

т. е. $c_{\boldsymbol{ab}}(\nu^{(k)})>c_{\boldsymbol{ab}}(\nu^{(k-1)})$ при каждом k. Следовательно,

$$c_{\boldsymbol{ab}}(\nu^{(0)})<c_{\boldsymbol{ab}}(\nu^{(n-1)})=c((1,2,\ldots,n))$$

для любой перестановки $\nu^{(0)}\neq(1,2,\ldots,n)$. Второе утверждение доказывается аналогично.

Опишем теперь множество перестановок, на котором достигается $\max\{c_{\boldsymbol{ab}}(\nu):\nu\in S_n\backslash\{(1,2,\ldots,n)\}\}$.

Л е м м а 4. *Если выполнены условия* (14), *то величина* $c_{\boldsymbol{ab}}(\nu)$ *на множестве перестановок* $\{\nu\in S_n:\nu_1=r\}$ *максимальна при*

$$\nu=(r,\varepsilon(N_n\backslash\{r\})). \tag{15}$$

При тех же условиях $\max\{c_{\boldsymbol{ab}}(\nu):\nu\in S_n\backslash\{(1,2,\ldots,n)\}\}$ *достигается на одной из транспозиций*

$$\nu^{(i)}=(\varepsilon(N_{i-1}),i+1,i,\varepsilon(N_n\backslash N_{i+1})),\quad i=1,\ldots,n-1.$$

Д о к а з а т е л ь с т в о. Первая часть утверждения непосредственно следует из леммы 3, так как если $\nu_1=r$, то

$$c_{\boldsymbol{ab}}(\nu)=a_1b_r+c_{(\boldsymbol{a}\backslash\{a_1\})(\boldsymbol{b}\backslash\{b_r\})}(\nu_2,\ldots,\nu_n),$$

где $(\nu_2,\ldots,\nu_n)$ — перестановка элементов множества $\{1,\ldots,n\}\backslash\{r\}$. Остается заметить, что в силу (14)

$$a_2>\ldots>a_n\quad\text{и}\quad b_1>\ldots>b_{r-1}>b_{r+1}>\ldots>b_n,$$

и применить лемму 3.

Для доказательства второй части леммы заметим, что

$$S_n\backslash\{(1,2,\ldots,n)\}=\bigcup_{i=1}^{n-1}\{\nu\in S_n:\nu_j=j\ (1\leqslant j<i),\ \nu_i>i\}$$

и поэтому согласно уже доказанному

$$\max_{\nu\in S_n\setminus(1,2,\ldots,n)} c_{\boldsymbol{ab}}(\nu)=\max_{1\leqslant i<n}\max_{\substack{\nu:\\ \nu_j=j\ (1\leqslant j<i),\nu_i>i}} c_{\boldsymbol{ab}}(\nu)=$$

$$=\max\left\{\max_{2\leqslant r<n} c_{\boldsymbol{ab}}((r,\varepsilon(N_n\setminus\{r\}))),\right.$$

$$\left.\max_{2\leqslant i<n}\max_{i<r\leqslant n} c_{\boldsymbol{ab}}\left((\varepsilon(N_{i-1}),r,\varepsilon(N_n\setminus(N_{i-1}\cup\{r\})))\right)\right\}. \quad (16)$$

Остается доказать, что если выполнены условия (14), то

$$\max_{i<r\leqslant n} c_{\boldsymbol{ab}}((1,\ldots,i-1,r,i,\ldots,r-1,r+1,\ldots,n))=c_{\boldsymbol{ab}}(\nu^{(i)}). \quad (17)$$

Достаточно рассмотреть случай $i=1$, т. е. проверить, что

$$\max_{1<r\leqslant n} c_{\boldsymbol{ab}}((r,\varepsilon(N_n\setminus\{r\})))=c_{\boldsymbol{ab}}((2,\varepsilon(N_n\setminus\{2\}))). \quad (18)$$

При любом $r=2,\ldots,n-1$ имеем:

$$c_{\boldsymbol{ab}}((r,\varepsilon(N_n\setminus\{r\})))-c_{\boldsymbol{ab}}((r+1,\varepsilon(N_n\setminus\{r+1\})))=$$

$$=a_1b_r+\sum_{j=2}^{r}a_jb_{j-1}+\sum_{j=r+1}^{n}a_jb_j-\left(a_1b_{r+1}+\sum_{j=2}^{r+1}a_jb_{j-1}+\sum_{j=r+2}^{n}a_jb_j\right)=$$

$$=a_1(b_r-b_{r+1})+a_{r+1}(b_{r+1}-b_r)=(a_1-a_{r+1})(b_r-b_{r+1})>0.$$

Тем самым (18) и (17) доказаны. Упрощая правую часть (16), с помощью (17), получаем второе утверждение леммы 4.

Чтобы сделать запись формул более компактной, будем для любого набора чисел $d=(d_1,d_2,\ldots,d_m)$ обозначать его отрезок $(d_i,d_{i+1},\ldots,d_j)$ символом d_i^j (при $i<j$); тем же символом будем обозначать и множество $\{d_i,d_{i+1},\ldots,d_j\}$.

Лемма 5. *Если выполнены условия* (14), $i\in\{1,\ldots,n-1\}$, *то для любой перестановки* $\sigma=(\sigma_1,\ldots,\sigma_n)=(\sigma_1^i,\varepsilon(N_n\setminus\sigma_1^i))\in S_n$

$$\max_{\nu\in S_n:\,\nu_1^i=\sigma_1^i} c_{\boldsymbol{ab}}(\nu)=c_{\boldsymbol{ab}}(\sigma), \quad (19)$$

и если $\theta_j(\sigma)=\min\{k:\sigma_j+1\leqslant k\leqslant n,\ k\notin\{\sigma_1,\ldots,\sigma_{j-1}\}\}$ (*при условиях леммы* $\theta_j(\sigma)=\sigma_{j+1}$, *если* $j>i$), *то*

$$\max_{\nu\in S_n\setminus\{\sigma\}:\,\nu_1^{i-1}=\sigma_1^{i-1},\nu_i\geqslant\sigma_i} c_{\boldsymbol{ab}}(\nu)=\max_{i-1\leqslant j<n} c_{\boldsymbol{ab}}((\sigma_1^j,\theta_j(\sigma),\varepsilon(N_n\setminus(\sigma_1^j\cup\{\theta_j(\sigma)\})))). \quad (20)$$

Равенство (19) непосредственно следует из леммы 3 и (15), а равенство (20) — из леммы 4.

Лемма 5 является обоснованием следующего алгоритма построения начальных отрезков ряда (13); он отличается от описания алгоритма, приведенного в начале работы, конкретным указанием способа преобразования границ D_t множеств Σ_t.

Построение начальных отрезков вариационного ряда (13).

1. Положить $t=1$, $D_1=\{(1,2,\ldots,n)\}$, $C_1=\{c_{\boldsymbol{ab}}((1,\ldots,n))\}$.

2. Найти $\nu^{(t)}=(\nu_{t1},\ldots,\nu_{tn})=\max\{c_{\boldsymbol{ab}}(\sigma):\sigma\in D_t\}$.

3. Если заданное число членов ряда (13) построено, то закончить работу, в противном случае перейти к п. 4.

4. Найти максимальное такое i, что

$$\nu^{(t)} = (\nu_{t1}, \ldots, \nu_{ti}, \varepsilon(\{\nu_{t,i+1}, \ldots, \nu_{tn}\})).$$

Если $i = n-1$, то положить

$$D_{t+1} = D_t \backslash \{\nu^{(t)}\} \quad \text{и} \quad D_{t+1} = C_t \backslash \{c_{\boldsymbol{ab}}(\nu^{(t)})\},$$

в противном случае положить

$$D_{t+1} = (D_t \backslash \{\nu^{(t)}\}) \cup \left(\bigcup_{j=i-1}^{n-1} \{(\sigma_1^j, \theta_j(\sigma), \varepsilon(N_n \backslash \{\sigma_1^j, \theta_j(\sigma)\}))\} \right)$$

и соответственно преобразовать множество C_t в C_{t+1}, обеспечивая возможность быстрого поиска его максимального элемента.

5. Увеличить t на 1 и перейти к п. 2.

Оценим вычислительную сложность алгоритма. При каждом значении t множество D_t увеличивается менее, чем на n элементов, поэтому на t-м шаге потребуется затратить $O(n)$ арифметических операций для вычисления значений $c_{\boldsymbol{ab}}(\sigma)$ и не более $O(nt)$ операций для переупорядочения множества C_{t+1}. Следовательно, общее число операций при построении первых T членов ряда (13) не превзойдет $O(nT^2)$.

СПИСОК ЛИТЕРАТУРЫ

1. Балакин Г. В. Алгоритм нахождения множества наименьшей мощности, содержащего истинное решение с заданной вероятностью // Труды по дискретной математике. — Т. 7. — М.: Физматлит. — 2003. — С. 7–21.
2. Кнут Д. Искусство программирования для ЭВМ. Т. 1. — М.: Мир, 1966.
3. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. — М.: Мир, 1973.
4. Харди Г. Г., Литтльвуд Д. Е., Полиа Г. Неравенства. — М.: Гос. изд. иностр. лит., 1948.
5. Hoeffding W. A combinatorial limit theorem // Ann. Math. Statist. — 1951. — V. 22. — P. 558–566.
6. Robinson J. A converse to a combinatorial limit theorem // Ann. Math. Statist. — 1972. — V. 43, № 6. — P. 2053–2057.
7. Wald A., Wolfowitz J. Statistical tests based on permutations of the observations // Ann. Math. Statist. — 1944. — V. 15. — P. 358–372.

Поступило в редакцию 7 VIII 2007