ние σ: $F\to\mathfrak{R}$, удовлетворяющее помимо условий 1)–2) также и такому: если для профиля $f\in F$ предпочтения $f(v)$ всех участников $v\in V$ являются слабыми упорядочениями, то и соответствующее групповое предпочтение $\sigma(f)$ – слабое упорядочение. В такой модели группового выбора соответствие между множеством агрегирующих функций Эрроу и множеством ультрафильтров булевой алгебры коалиций уже взаимно-однозначно (теорема 1 в [6]).

ЛИТЕРАТУРА

1. *Armstrong T. E.* Arrow's Theorem with Restricted Coalition Algebras.— J. Math. Ec., 1980, v. 7, № 1.
2. *Arrow K.* Social Choice and Individual Value. N. Y., 1963.
3. *Kirman A., Sonderman D.* Arrow's Theorem, Many Agents and Invisible Dictators.— J. Ec. Th., 1972, v. 5, № 2.
4. *Тангян А. С.* Модели социального выбора с конечным и бесконечным числом участников. М.: ЦЭМИ АН СССР, 1979.
5. *Тангян А. С.* Агрегирование в модели социального выбора. М.: ЦЭМИ АН СССР, 1979.
6. *Тангян А. С.* Иерархическая модель группового выбора.– Экономика и матем. методы, 1980, т. XVI, вып. 3.
7. *Тангян А. С.* Переход к бесконечному числу участников в модели группового выбора.– Экономика и матем. методы, 1981, т. XVII, вып. 1.

Поступила в редакцию
29 IV 1980

МИНИМАКСНАЯ ЗАДАЧА О ТРИАНГУЛЯЦИИ МАТРИЦЫ

Бурдюк В. Я.

(Днепропетровск)

1. Постановка задачи. Заданы множество $N=\{1,\ldots,n\}$, вещественная $n\times n$-матрица $a=(a_{ij})=(a[i,j])$ и функционал

$$f=f(S,a)=\min_{1\leqslant p<q\leqslant n}\{a[S[p],S[q]]\},$$

где S – произвольная перестановка множества N.

Требуется среди $n!$ перестановок S найти перестановку $S_{\max}$, которая максимизирует функционал f. Другими словами, нужно, пользуясь одинаковыми переупорядочениями строк и столбцов квадратной матрицы, максимизировать минимальный элемент выше ее главной диагонали. Эта задача даст возможность решить и задачу минимизации (при помощи одинакового переупорядочения строк и столбцов квадратной матрицы) максимального элемента над ее главной диагональю, ибо

$$\max_{p<q}\{a[S[p],S[q]]\}=-\min_{p<q}\{-a[S[p],S[q]]\}.$$

Ниже предложен алгоритм построения при любом $\varepsilon>0$ такой перестановки S_0, что $|f(S_0)-f(S_{\max})|\leqslant\varepsilon$. Количество элементарных действий (сравнений, сложений чисел) в этом алгоритме не больше $(1+\log_2(\beta/\varepsilon))c_1n^2$, а объем дополнительной памяти – (n^2+2n) ячеек. Заметим, что если, например, все числа a_{ij} имеют два знака после запятой, то по алгоритму получим S_0, равное $S_{\max}$, задав $\varepsilon=0{,}009$.

Известно, что обычная задача о триангуляции матрицы (см. [1]), т. е. задача максимизации функционала

$$f_1=f_1(S,a)=\sum_{p<q}a[S[p],S[q]],$$

решается значительно сложнее.

2. Алгоритм. Считая заданными $\varepsilon>0$ и $n\times n$-матрицу $a=(a_{ij})=(a[i,j])$, выполняем следующие действия.

Шаг 1. Задаем $n\times n$-матрицу $d=(d_{ij})=(d[i,j])$, где $d_{ii}=0$, $d_{ij}=1$, $i\neq j$. Просматривая элементы матрицы a, расположенные ниже (выше) главной диагонали, находим среди них наибольший M_- (соответственно M_+) и наименьший m_- (соответственно m_+). Полагаем $M:=\max(M_-,M_+)$, $m:=\max(m_-,m_+)$. Если $m=m_+$, то $S_0:=\langle 1,\ldots,n\rangle$, иначе $S_0:=\langle n,n-1,\ldots,1\rangle$.

Шаг 2. Вычисляем $\delta:=M-m$. Если $\delta\leqslant\varepsilon$, то переходим к шагу 10.

Шаг 3. Вычисляем $t:=m+\delta/2$.

Шаг 4. Для любых $i\neq j$, если $d[i, j]=1\wedge a[i, j]\leqslant t$, то $d[i, j]:=2$.

Шаг 5. Находим вектор B, компонента $B[i]$ которого, $i=1,\ldots,n$, равна количеству единиц в i-й строке матрицы d. Полагаем $k:=0$.

Шаг 6. Для $p:=1$ с шагом 1 до n выполняем следующие действия. Полагаем $r:=n-p$. Ищем наименьшее j такое, что $B[j]=r$. Если указанного j нет, то переходим к шагу 7. Полагаем $S[p]:=j$; $B[j]:=-1$. Для $i:=1$ с шагом 1 до n выполняем: если $d[i, j]=1$, то $B[i]:=B[i]-1$. Полагаем $k:=k+1$.

Шаг 7. Если $k<n$, то переходим к шагу 9.

Шаг 8. Для любых $i\neq j$, если $d[i, j]=2$, то полагаем $d[i, j]:=0$. Для $q:=1$ с шагом 1 до n выполняем $S_0[q]:=S[q]$. Находим $m:=\min\limits_{p<q}\{a[S_0[p], S_0[q]]\}$. Возвращаемся к шагу 2.

Шаг 9. Полагаем $M:=t$. Для любых $i\neq j$, если $d[i, j]=2$, то полагаем $d[i, j]:=1$. Возвращаемся к шагу 2.

Шаг 10. Конец. Найдены S_0 и $f(S_0)=m$.

Рассмотрим пример. Пусть $\varepsilon=0{,}9$, $n=4$, $(a_{1j})=(0,\ 10,\ 1,\ 9)$, $(a_{2j})=(2,\ 0,\ 8,\ 13)$, $(a_{3j})=(14,\ 12,\ 0,\ 8)$, $(a_{4j})=(3,\ 9,\ 20,\ 0)$. Тогда $M_-=20$, $m_-=2$, $M_+=13$, $m_+=1$, $M=20$, $m=2$, $S_0=\langle 4,\ 3,\ 2,\ 1\rangle$, $\delta=18$, $t=11$. Затем следуют шаги 6, 7, 9, 2, 3, в результате чего получаем $M=11$, $\delta=9$, $t=6{,}5$. После шагов 4–8 имеем

$$d=\begin{pmatrix}0&1&0&1\\0&0&1&1\\1&1&0&1\\0&1&1&0\end{pmatrix},\qquad S_0=\langle 3,\ 1,\ 2,\ 4\rangle,\qquad m=8.$$

Затем $\delta=3$, $t=9{,}5$, $M=9{,}5$, $\delta=1{,}5$, $t=8{,}75$, $M=8{,}75$, $\delta=0{,}75$, т. е. $\delta<\varepsilon$. В итоге $S_0=\langle 3,\ 1,\ 2,\ 4\rangle$, $S_0=S_{\max}$.

3. Обоснование алгоритма и оценка сложности. Идея алгоритма состоит в том, чтобы, запретив появляться выше главной диагонали тем элементам a_{ij} матрицы a, которые не превосходят заданного уровня запрета t, проверить затем, существует ли перестановка S, обеспечивающая заполнение мест над главной диагональю незапрещенными элементами. Если S существует, то уровень запрета увеличивается. Заметим, что назначение шагов 1, 2, 3 алгоритма легко уяснить. На шаге 4 как раз и происходит запрет элементов матрицы a увеличением единичных элементов вспомогательной матрицы d, причем если такой запрет оказался успешным, то на шаге 8 этот запрет производится окончательно, ибо элемент a_{ij} не запрещен, если только $d_{ij}=1$. Шаги 5 и 6 предназначены для построения перестановки S такой, что в матрице $(d[S[p],\ S[q]])$ выше главной диагонали находятся только единицы, а также для указания, что делать, если S не существует (в этом случае шаг 7 отправляет нас на шаг 9, в результате чего уровень запрета понижается). В случае построения S происходит переприсваивание $S_0:=S$ и вычисление $m=f(S_0)$ на шаге 8. Отметим, что если S построено, то это S является регулярным упорядочением вершин орграфа, заданного (см. [2]) матрицей смежности (d_{ij}).

Из структуры алгоритма непосредственно видно, что количество перебираемых перестановок S меньше числа $1+\log_2(\beta/\varepsilon)$, где $\varepsilon>0$ – точность решения, а $\beta=\max\limits_{i\neq j}a_{ij}-\min\limits_{i\neq j}a_{ij}$. Можно подсчитать, что сложность построения одной перестановки S не превосходит c_1n^2 элементарных действий. Следовательно, количество элементарных действий в этом алгоритме не превосходит числа $(1+\log_2(\beta/\varepsilon))c_1n^2$.

4. Модификация алгоритма. Используя предыдущий алгоритм, опишем более «естественный» алгоритм построения $S_{\max}$ для f, а затем покажем, что модифицированный алгоритм может потребовать $c_2n^2n(n-1)/2$ элементарных действий, где $c_2<c_1$.

Считаем заданной $n\times n$-матрицу $a=(a_{ij})=(a[i, j])$.

Шаг 1. Задаем $n\times n$-матрицу $d=(d_{ij})$, где $d_{ii}=0$, $d_{ij}=1$, $i\neq j$.

Шаг 2. Задаем перестановку S_0, полагая $S_0[i]:=i$ для $i=1,\ldots,n$.

Шаг 3. Находим $m:=\min\limits_{p<q}\{a[S_0[p], S_0[q]]\}$.

Шаг 4. Для любых $i\neq j$, если $d[i, j]=1\wedge a[i, j]\leqslant m$, то $d[i, j]:=0$.

Шаги 5, 6 и 7 такие же, как и в предыдущем алгоритме.

Шаг 8. Полагаем $S_0:=S$. Возвращаемся к шагу 3.

Шаг 9. Конец. Найденное S_0 есть $S_{\max}$, $m=f(S_{\max})$.

В модифицированном алгоритме сложность построения одной перестановки S не превосходит числа c_2n^2, где $c_2<c_1$. Но следующий пример показывает, что количество перебираемых перестановок S при этом может достигать максимального значения, т. е. равняться $n(n-1)/2$. Действительно, пусть числа $1,\ldots,n(n-1)/2$ размещены над главной диагональю матрицы a так, что по строкам слева направо и по столбцам сверху вниз числа a_{ij} возрастают, т. е. $a_{12}=1$, $a_{13}=2,\ldots,a_{1n}=n-1$, $a_{23}=n$

$a_{24}=n+1, \ldots, a_{n-1,\,n}=n(n-1)/2$. Далее, пусть $a_{ij}>n(n-1)/2$ при $i>j$. Тогда при k-м повторении шага 3 нового алгоритма будем иметь $m:=k$, где $k=1,\ldots,n(n-1)/2$.

В заключение автор выражает признательность В. П. Козыреву за весьма полезное обсуждение результатов данной работы.

ЛИТЕРАТУРА

1. *Баранов Э. Ф.* О последовательности народнохозяйственного производственного процесса.— В кн. Проблемы планирования и прогнозирования. М.: Наука, 1974.
2. *Бурдюк В. Я.* Регулярные g-упорядочения и функции Смита.— Кибернетика, 1975, № 2.

Поступила в редакцию
9 I 1978

О ПОВЕДЕНИИ НАСЕЛЕНИЯ В СОЦИАЛЬНО-ПРОИЗВОДСТВЕННОЙ МОДЕЛИ

Бородкин Ф. М.

(Новосибирск)

В [1] приведена некоторая задача линейного программирования, решение которой эквивалентно решению дифференциальных уравнений

$$\sum_{i=1}^{3}\alpha_i\dot{F}_i=K, \tag{1}$$

$$\sum_{i=1}^{3}s_i\dot{F}_i=\dot{F}_3, \tag{2}$$

$$\sum_{i=1}^{3}\mu_iF_i=hP, \tag{3}$$

$$\dot{P}=-\eta(K_1-K_2), \tag{4}$$

$$K_1=\alpha_1\dot{F}_1+\alpha_3\dot{F}_3, \tag{5}$$

$$K_1+K_2=K, \tag{6}$$

где F_i, $i=1, 2, 3$ – основные фонды сельскохозяйственного производства, строительства и непроизводственной сферы соответственно; K, K_1, K_2 – инвестиции; P – численность населения сельской части региона; α_i, s_i, μ_i, η – положительные коэффициенты. В общем случае все переменные и коэффициенты зависят от времени. Точка над переменной означает производную по времени. Если первые три уравнения не вызывают особых сомнений, поскольку они являются традиционно балансовыми, то (4) было подвергнуто резкой критике за форму, обоснование и интерпретацию. Эта критика стимулировала размышления по поводу данного уравнения и привела к интересным, на наш взгляд, результатам, которыми я и хотел поделиться с читателями. Это тем более необходимо, что данная система уравнений используется в прогнозировании развития сибирского села. Отмечу сразу же, что критика по существу своему была верной.

Рассмотрим некоторый регион с двумя сферами (множествами отраслей) – производственной и непроизводственной, основные фонды которых обозначим соответственно через F_1 и F_2, а инвестиции $K_1=\alpha_1\dot{F}_1$, $K_2=\alpha_2\dot{F}_2$; $K=K_1+K_2$. Уравнение (3) тогда запишется в виде

$$\mu_1F_1+\mu_2F_2=hP. \tag{7}$$

Обозначим численность рабочей силы, используемой в производственной сфере и измеренной в каких-либо подходящих единицах, через $L=\mu_1F_1$.