УДК 531.8

*В. И. РУБЛИНЕЦКИЙ, И. И. ТАРТАКОВСКИЙ*

# КРИТЕРИЙ СУЩЕСТВОВАНИЯ И АЛГОРИТМ НАХОЖДЕНИЯ РЕГУЛЯРНЫХ РЕАЛИЗАЦИЙ ПОХОДКИ МНОГОНОГОЙ ШАГАЮЩЕЙ СИСТЕМЫ

В последние годы большое внимание уделяется изучени походок естественных и искусственных шагающих систем [1, 2] Среди различных характеристик походок одной из важнейши является регулярность. При регулярной походке режимы рабо ты всех ног идентичны, что позволяет существенным образом упростить управление их движением. Такие походки, как пра вило, используют при передвижении по ровной местности жи вотные. Они кажутся предпочтительными в этих условиях и для шагающих автоматов.

При задании походки с помощью последовательности состояний возникает вопрос об установлении возможности регулярной реализации походки [2]. Его решение связывается в общем случае с некоторой задачей линейного программирования и в массовых расчетах по анализу походок не эффективно. Ниже рассматривается существенно использующий специфику исходной задачи новый критерий регулярной реализуемости походки и основанный на нем алгоритм поиска таких реализаций. Показано, что для $N$-ногой шагающей системы алгоритм требует порядка $N^3$ элементарных операций.

1. *Постановка задачи.* Введем основные определения, следуя [2, 3]. Состоянием $N$-ногого шагающего автомата называется вектор $(q_1, ..., q_N)$, где $q_j=0$, если нога с номером $j$ находится в фазе опоры, и $q_j=1$, если она находится в фазе переноса. Походке отвечает периодическая последовательность состояний, такая что состояние каждой ноги изменяется за цикл походки один раз от 0 к 1 и один раз от 1 к 0. Эту последовательность удобно представить в виде матрицы $G$ размера $M\times N$, строки которой соответствуют $M$ последовательным состояниям автомата в одном цикле походки. Конкретная реализация походки полностью задается парой $\{G, t\}$, где $t=(t_1, ..., t_M)$ — вектор расписания походки; $t_i$ — длительность состояния, отвечающего $i$-й строке матрицы $G$.

Характеристикой режима $\beta_j$ ноги с номером $j$ называется отношение времени $\tau_j$, в течение которого эта нога находится в пределах одного цикла в фазе опоры, к времени цикла $T=\sum_{i=1}^{M} t_i$. Походка регулярна, если $\beta_1=\beta_2=\ldots=\beta_N$.

Вектор $\beta=(\beta_1, \ldots, \beta_N)$ можно вычислить по формуле

$$\beta=e-\frac{1}{T}tG. \qquad (1)$$

где $e$ — $N$-мерный вектор, все координаты которого равны единице.

Матрица $G$ называется регулярной, если существует такое расписание $t$, что походка $\{G, t\}$ регулярна. Из формулы (1) сразу следует, что матрица $G$ регулярна тогда и только тогда, когда существует положительное решение $t^0$ системы $N$ уравнений

$$tG=e. \qquad (2)$$

Вектор $v$ называется положительным (обозначение $v>0$), если все его компоненты положительны, и полуположительным ($v\geqslant 0$), если все его компоненты неотрицательны, но хотя бы одна из них положительна.

Таким образом, критерий регулярности матрицы походки $G$ эквивалентен критерию существования положительных решений матричного уравнения (2).

Известные результаты, относящиеся к рассматриваемому вопросу, сводятся к следующим трем утверждениям. Введем *условие* $C_1$: матрица $G$ содержит столбцы $g^i$ и $g^j$, такие что $g^i - g^j \geqslant 0$ (вектор-столбец $g^i$ мажорирует вектор-столбец $g^j$).

Очевидно *утверждение 1*: если выполнено условие $C_1$, то уравнение (2) не имеет положительных решений. Матрицы $G$, для которых условие $C_1$ не выполняется, называются нормальными [3]. Таким образом, нормальность матрицы является необходимым условием ее регулярности.

Легко проверяется справедливость *утверждения 2*, дающего достаточное условие регулярности матрицы для одного частного случая: если все столбцы матрицы $G$ имеют одинаковое число единиц, то она регулярна.

В общем же случае известно только следующее утверждение, позволяющее использовать для решения вопроса о существовании положительного решения системы (2) методы линейного программирования.

*Утверждение 3* [2]. Система (2) имеет положительное решение тогда и только тогда, когда задача

$$T = \sum_{i=1}^{M} t_i \to \max \tag{3}$$

с ограничениями-равенствами (2) и ограничениями-неравенствами $t_i > 0$ $(i = 1, \ldots, M)$ имеет допустимые решения.

Целевую функцию в задаче (3) можно, конечно, заменить любой другой линейной функцией переменных $t_i$.

Заметим, что ни одно из приведенных известных утверждений не использует специфической особенности матрицы $G$, связанной с ее происхождением — в каждом из $N$ столбцов матрицы все единицы стоят подряд, единым блоком, если считать, что первая строка циклически следует за последней.

2. *Критерий существования положительных решений.* Прежде всего несколько упростим задачу. Матрица походки $G$ может, вообще говоря, иметь строки, состоящие из одних нулей (такая строка отвечает состоянию, в котором все ноги находятся в фазе опоры) или из одних единиц (состояние прыжка — все ноги в фазе переноса). Ясно, что возможность регулярной реализации походки с такой матрицей не зависит от длительности указанных состояний. Поэтому указанные строки можно вычеркнуть из матрицы $G$. Если в матрице имеются одинаковые столбцы, то можно оставить только один из них, так как таким столбцам отвечают одинаковые уравнения в системе (2).

В дальнейшем будем считать, что рассмотренные упрощения, в результате которых матрица $G$ превращается в новую матрицу $B$, уже выполнены и вместо (2) следует анализировать уравнение

$$xB = e. \tag{4}$$

Матрица $B$ имеет размеры $m \times n$ ($m \leqslant M$, $n \leqslant N$), $x$ — $m$-мерный, а $e$ — $n$-мерный векторы.

Так как все матрицы, получающиеся из данной матрицы походки циклической перестановкой строк, отвечают одной и той же походке, будем для матрицы $G$ использовать каноническую по строкам [1] форму, в которой первая строка начинается нулем, а последняя — единицей. Если матрица $G$ является нормальной, то в полученной из нее матрице $B$ блоки нулей во всех столбцах начинаются в различных строках. Но тогда столбцы можно переставить так, чтобы они были упорядочены по возрастанию указанных строк. Ясно, что перестановка столбцов соответствует перенумерации ног шагающей системы и не влияет на свойство регулярности походки.

Рассмотрим примеры двух матриц, приведенных в результате описанных преобразований к стандартной, канонической по строкам и столбцам [1] форме:

$$B_1 = \begin{bmatrix} 0&1&1&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&1&1&0 \\ 0&0&0&0&1&1 \\ 1&0&0&0&1&1 \\ 1&0&0&0&0&1 \\ 1&1&0&0&0&0 \\ 1&1&1&0&0&0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0&1&1&0&0&0 \\ 0&0&1&1&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&1&1&0 \\ 0&0&0&0&1&1 \\ 1&0&0&0&0&1 \\ 1&1&0&0&0&1 \\ 1&1&0&0&0&0 \end{bmatrix}. \qquad (5)$$

Легко видеть, что и в столбцах, и в строках матриц стандартной формы все нули или (и) все единицы располагаются едиными блоками. Будем считать далее, что матрица $B$ в уравнении (4) имеет стандартную форму, и выполним следующее эквивалентное преобразование: вместо уравнения (4) или в векторной записи

$$xb^j = 1, \ (j = 1, \ \ldots, \ n)$$

рассмотрим систему уравнений (6):

$$xb^1 = 1, \qquad (6.1)$$

$$x(b^j - b^{j+1}) = 0, \ \ (j = 1, \ \ldots, \ n-1); \qquad (6.2)$$

$$x(b^n - b^1) = 0 \qquad (6.3)$$

(уравнение (6.3) вытекает из (6.2) и введено для технического удобства).

Система (6) имеет положительное решение тогда и только тогда, когда его имеет подсистема (6.2). Собственно, подсистема (6.2) задает соотношение переменных, а уравнение (6.1) — масштаб. В одну сторону это утверждение тривиально: если подсистема (6.2) не имеет положительного решения, то его не имеет и вся система. Наоборот, пусть (6.2) имеет положительное решение $\bar{x}^\circ$. Поскольку $\bar{x}^\circ > 0$ и $b^1 \geqslant 0$, то $\bar{x}^\circ b^1 = \tau > 0$. Положим $x = \bar{x}/\tau$; $x$ очевидно удовлетворяет системе (6).

Обозначим через $A$ матрицу системы уравнений (6.2), (6.3). Матрицы $A_1$ и $A_2$ получены из приведенных выше матриц $B_1$ и $B_2$. Для простоты здесь и ниже обозначения + и — употреблены вместо +1 и —1.

$$A_1 = \begin{bmatrix} - & 0 & + & 0 & 0 & 0 \\ 0 & - & 0 & + & 0 & 0 \\ 0 & 0 & - & 0 & + & 0 \\ 0 & 0 & 0 & - & 0 & + \\ + & 0 & 0 & - & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 \\ 0 & + & 0 & 0 & 0 & - \\ 0 & 0 & + & 0 & 0 & - \end{bmatrix}, \quad A_2 = \begin{bmatrix} - & 0 & + & 0 & 0 & 0 \\ 0 & - & 0 & + & 0 & 0 \\ 0 & 0 & - & + & 0 & 0 \\ 0 & 0 & - & 0 & + & 0 \\ 0 & 0 & 0 & - & 0 & + \\ + & 0 & 0 & 0 & - & 0 \\ 0 & + & 0 & 0 & - & 0 \\ 0 & 0 & + & 0 & 0 & - \end{bmatrix}. \tag{7}$$

В связи с блочным расположением нулей и единиц в строках матрицы $B$ каждая строка матрицы $A$ имеет только два ненулевых элемента: один, равный +1, а другой — —1. Таким образом, задача сведена к выяснению условий, при которых уравнение

$$xA = 0 \tag{8}$$

имеет положительное решение.

В теории линейных неравенств известен следующий результат, касающийся произвольной матрицы $A$ [4, с. 79].

*Утверждение 4.* Имеет место ровно одна из следующих альтернатив: либо уравнение (8) имеет положительное решение, либо разрешимо неравенство

$$Ay \geqslant 0. \tag{9}$$

Сформулируем следующее условие $C_2$: существует набор индексов $P = (j_1, \ldots, j_k) \subset (1, \ldots, n)$ такой, что

$$\sum_{j \in P} a^j \geqslant 0. \tag{10}$$

Если $k$ — число индексов в наборе $P$ — равно 1, то получается условие $C_1$. Ясно также, что $k \leqslant n-1$. Докажем две леммы.

***Лемма 1.*** *Если выполняется условие $C_2$, то разрешимо неравенство (9).*

Доказательство. Построим $y$ по следующему правилу:

$$y_j = \begin{cases} 1, & \text{если } j \in P; \\ 0, & \text{если } j \,\bar{\in}\, P. \end{cases}$$

Доказательство завершается подстановкой $y$ в неравенство (9) и использованием условия (10).

***Лемма. 2.*** *Если условие $C_2$ не выполняется, то неравенство (9) неразрешимо.*

Доказательство. Возьмем произвольный вектор $y$; пусть в нем имеется $r$ положительных, $s$ отрицательных и $t$ нулевых компонент. Сгруппируем положительные компоненты в начале вектора, а нулевые — в конце. Соответственно переставим столбцы матрицы (рис. 1).

Заметим, что в каждом столбце матрицы $A$ имеется хотя бы один минус, иначе было бы верно $C_1$.

Рассмотрим подматрицу, заключенную в рамку на рис. 1. Если в рамке в строке $a_i$ стоит минус, а однострочный плюс стоит вне рамки, то $a_i y<0$, т. е. неравенство (9) неразрешимо. Если, наоборот, в рамке плюс, а вне рамки — минус, то сумма столбцов в рамке полуположительная, т. е. верно $C_2$. Поэтому в строке в рамке стоит и плюс, и минус (пусть число таких строк равно $l$ и они сгруппированы сверху) либо нет ни плюса, ни минуса.

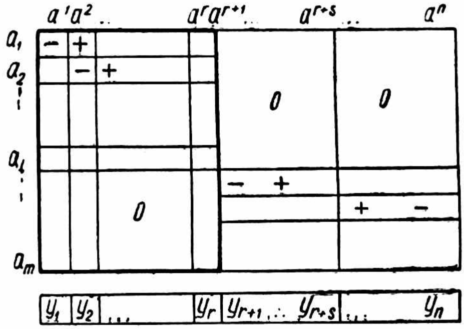


Рис. 1. Положительные и нулевые компоненты вектора

Если $y_1=y_2=\ldots=y_r$, то, очевидно, для $i=1, \ldots, l$ имеют место равенства $a_i y=0$, а для того, чтобы было $Ay\geqslant 0$, нужно получить хоть одну положительную компоненту. Пусть числа $y_1, \ldots, y_r$ не все равны и $y_1=\max\limits_{1<j<r} y_j$. В столбце $a^1$ имеется минус; пусть для определенности он стоит в строке $a_1$, как на рис. 1. В рамке стоит однострочный плюс, скажем, в столбце $a^2$. Поскольку $y_1$ максимален, то либо $y_2<y_1$, либо $y_2=y_1$. В первом случае $a_1 y<0$ и все доказано, во втором — $a_1 y=0$. Продолжая в том же духе, мы видим, что всякое вовлеченное в рассуждение $y_j$ должно равняться $y_1$. Поэтому либо будет доказано равенство всех чисел множества $\{y_1, \ldots, y_r\}$, либо оно разобьется на несколько подмножеств равных между собой чисел. В любом случае получим равенства $a_i y=0$ для $i=1, \ldots, l$.

Аналогичное рассуждение (с заменой $>$ на $<$ и max на min) проводится для группы отрицательных компонент вектора $y$. Для нулевой части вектора рассуждение тривиально. Лемма 2 доказана.

Условие $C_2$ для матрицы $B$ превращается в условие $C$: существует множество индексов $P=(j_1, \ldots, j_k)$ и множество $R=(i_1, \ldots, i_k)$, такие что

$$\sum_{j\in P} b^j-\sum_{i\in R} b^i\geqslant 0.$$

Из доказанного выше следует

**Теорема 1.** *Имеет место ровно одна из следующих альтернатив: либо уравнение (4) имеет положительное решение, либо верно условие $C$.*

3. *Конструктивный критерий.* Проверка условия $C_2$, фигурирующего в леммах 1 и 2, «в лоб» требует экспоненциального числа операций, что в прикладном смысле неконструктивно. Пользуясь спецификой матрицы $A$, опишем алгоритм проверки условия $C_2$, имеющий кубическую сложность.

Итак, если верно $C_2$, то найдется множество $P$ номеров, $|P| \in \overline{1, n-1}$, такое что сумма столбцов с этими номерами образует полуположительный вектор. Пусть положительна $i$-я компонента этого вектора. В строке $a_i$ имеется ровно один плюс (в столбце $a^r$) и ровно один минус (в столбце $a^s$). Множество $P$, очевидно, должно обладать следующими свойствами:

(1) $r \in P$ (чтобы $i$-я компонента суммы была положительна);

(2) $s \overline{\in} P$ (чтобы плюс из $a_i$ не «скомпенсировался»);

(3) если в множество $P$ входит столбец с минусом в $l$-й строке, то в него входит и столбец с плюсом в этой строке (чтобы не осталось нескомпенсированных минусов).

Эти свойства позволяют построить множество $P$ следующим образом: сначала включить $r$ в $P$, затем включить в него столбцы с плюсами, компенсирующими минусы столбца $a^r$, и т. д. Если, в конечном счете, наберется множество $P$, где все минусы скомпенсированы однострочными плюсами и куда не входит $s$, то условие $C_2$ верно, иначе нужно перейти к проверке другого значения $i$, пока не будет найдено множество $P$, удовлетворяющее неравенству (10), или показано, что ни для какого $i$ не существует такого множества. В последнем случае условие $C_2$ не выполняется.

Для подсчета числа требуемых элементарных операций опишем алгоритм точнее. Пусть матрица $A$ представлена массивом Minus $[1:m]$ (Minus $(l)$ задает номер столбца, где в $l$-й строке матрицы $A$ стоит минус) и аналогичным массивом Plus $[1:m]$. Введем еще рабочие бинарные массивы Mi $[1:m]$ и Pl $[1:m]$, где единицами отмечаются соответственно нескомпенсированные минусы и плюсы текущего отобранного множества столбцов.

**Алгоритм 1**

1°. $i := 0$.

2°. . $i := i + 1$; если $i > m$, то перейти к 8°.

3°. $\mathrm{Pl} := 0$; $\mathrm{Mi} := 0$; $r := \mathrm{Plus}(i)$; $s := \mathrm{Minus}(i)$.

4°. Цикл по $l = \overline{1, m}$:

если $\mathrm{Minus}(l) = r$, то $\{\mathrm{Mi}(l) := 1 - \mathrm{Pl}(l);\ \mathrm{Pl}(l) := 0\}$;

если $\mathrm{Plus}(l) = r$, то $\{\mathrm{Pl}(l) := 1 - \mathrm{Mi}(l);\ \mathrm{Mi}(l) := 0\}$.

5°. $z := 0$; цикл по $l = \overline{1, m}$: если $\mathrm{Mi}(l) = 1$, то $\{z := l$; перейти к 7°$\}$.

6°. $C_2$ верно, конец.

7°. $r := \mathrm{Plus}(z)$; если $r \neq s$, то перейти к 4°, иначе — к 2°.

8°. $C_2$ неверно, конец.

Теперь легко подсчитать сложность алгоритма: шаг 2° задает цикл $\overline{1, m}$, шаг 4° — цикл $\overline{1, m}$ в цикле, а шаг 7° задает повторение шага 4° не более, чем $n$ раз. Всего получаем сложность порядка $m^2n$ или $N^3$, так как $n \leqslant N$, $m \leqslant M$, $M \leqslant 2N$ [2]. Сложность предварительного вычисления справочных массивов имеет порядок $mn$ и не повышает оценку сложности.

Теперь рассмотрим полученный результат в другом разрезе. Положим, что алгоритм кончает работу шагом 8°, когда $C_2$ неверно, т. е. уравнение (8) имеет положительное решение. Это означает, что для каждого $i$ итерация кончается на шаге 7° по условию $r=s$, т. е. что в некоторой строке $i=i_1$ в начале итерации был зафиксирован плюс (в столбце $a^r$); для компенсации какого-то из минусов столбца $a^r$ (скажем, этот минус стоял в строке $i_2$) был введен столбец с плюсом в этой строке, но он имел минус в $i_3$-й строке и т. д., пока компенсирующий плюс не был найден в столбце $a^s$ в строке $a_k$.

Другими словами, в ходе итерации был построен путь, обладающий следующими свойствами:

(1) путь состоит только из вертикальных (направленных по столбцу) и горизонтальных (направленных по строке) дуг;

(2) если дуга начинается плюсом, она кончается минусом, и наоборот;

(3) путь начинается вертикальной дугой в строке $a_{i_1}$ в столбце $a^r$;

(4) путь кончается плюсом горизонтальной дугой в некоторой строке $a_{i_k}$ в столбце $a^s$;

(5) если добавить к пути вертикальную дугу от последнего плюса в столбце $a^s$ до минуса в этом столбце в строке $a_{i_1}$ и еще горизонтальную дугу до начального плюса в строке $a_{i_1}$, то получится контур.

Контур, обладающий свойствами (1) — (5), назовем уравновешенным. Если надо дополнительно отметить, что этот контур имеет горизонтальную дугу в строке $i$, будем называть его уравновешенным $i$-контуром или, короче, $i$-контуром.

Очевидно, что уравновешенный контур содержит четное число дуг (горизонтальных столько же, сколько вертикальных) и что он однозначно описывается перечислением строк, где проходят его горизонтальные дуги; так, контур с $2k$ дугами задается множеством $Q=(i_1, i_2, ..., i_k)$.

Из предыдущих рассуждений следует

**Теорема 2:** *Уравнение $xA=0$ имеет положительное решение тогда и только тогда, когда для любого $i$ в матрице $A$ найдется $i$-контур, что проверяется алгоритмом 1 за $\sim n^3$ операций.*

Применяя алгоритм 1 к матрицам (7), получим для матрицы $A_1$ останов на шаге 6° после первой итерации, а для матрицы $A_2$ — на шаге 8°.

4. *Алгоритм нахождения положительного решения.* Мы опи-

шем здесь алгоритм 2, который не только проверяет, как алгоритм 1, существование положительного решения уравнения (8), но и строит его, если оно существует.

Идея алгоритма 2 такова. При $x=0$ уравнение (8) выполняется тривиально. Попытаемся построить $i$-контур, $i \in M=\{1, \ldots, m\}$. Если такого контура построить невозможно, то по теореме 2 уравнение (8) не имеет решения, конец. Пусть $i$-контур построен (как это сделать, обсудим позже); этот контур описывается множеством $Q=(i_1, i_2, \ldots, i_k)$ вошедших в него строк. Построим новый вектор $x'$ по следующему правилу:

$$x'_l = \begin{cases} x_l + 1, & \text{если } l \in Q; \quad (l = 1, \ldots, m). \\ x_l, & \text{если } l \notin Q. \end{cases} \tag{11}$$

Покажем, что $x'A = 0$ или в векторной форме

$$x'a^j = 0, \; j = 1, \ldots, n. \tag{12}$$

Действительно, если столбец $a^j$ таков, что в нем не содержится вертикальных дуг $i$-контура, то во всех строках с номерами из $Q$ стоят нули и каждая увеличенная на единицу компонента исчезает, умножаясь на нуль. Если же в столбце $a^j$ содержится одна или несколько вертикальных дуг $i$-контура, то плюсы и минусы входят парами, взаимно уничтожаясь друг с другом.

Переобозначим $x:=x'$. Теперь нулевые компоненты стоят в $x$ только на местах, принадлежащих множеству $M:=M/Q$. Итерируя этот алгоритм по $i$, $i \in M$, строим положительный вектор $x$ или убеждаемся, что такого вектора не существует.

Теперь обсудим, как строить $i$-контур. Если применять способ, использованный в алгоритме 1, то трудно понять, от какого именно минуса введенного столбца нужно вести горизонтальную дугу, чтобы получить $i$-контур. Мы применим другую технику.

Пусть в ходе построения $i$-контура обозревается минус в элементе $a_{lj}$. От этого минуса имеется единственное продолжение по строке $l$ к плюсу в столбце $j'=\mathrm{Plus}(l)$. Пусть в $j'$-м столбце минусы стоят в строках с номерами $l_1, l_2, \ldots, l_\lambda$. Значит, можно (за $\sim m$ операций) составить строчку

$$l_1, l_2, \ldots, l_\lambda \tag{13}$$

номеров строк (в порядке возрастания), где стоят минусы, могущие следовать в $i$-контуре за минусом в $l$-й строке. Строчки вида (13) для $l=\overline{1, m}$ можно рассматривать как таблицу следования $T$, задающую орграф Г, ассоциированный с матрицей $A$. Нахождение контура, содержащего вершину $l$ в графе Г, эквивалентно нахождению $i$-контура в матрице $A$.

Проверка наличия контуров в орграфе и их эффективное построение изучалось в теории графов (см., например, [5]). Однако наша задача попроще — для каждой вершины найти хоть один контур, ее содержащий, и мы предложим для нее более простой алгоритм 3.

Пусть $T$ — таблица следования; $u$, $v$ $[1:m]$ — рабочие массивы; $Q$ — массив, содержащий номера вершин контура. Пусть надо построить контур, содержащий вершину $i$.

### Алгоритм 3

1°. $u := 0$; $v := 0$; $u(i) := 1$.

2°. Найти в массиве $u$ номер $l$ первой единицы. Если нет единиц, то перейти к 6°.

3° Извлечь из $T$ набор вершин, следующих за $l$-й: $l_1$, $l_2$, $\ldots$, $l_\lambda$.

4°. Цикл по $j = l_1, \ldots, l_\lambda$: если $j = i$, то перейти к 5°, иначе, если $v(j) = 0$, то $\{v(j) := l;\ u(j) := 1\}$.

После цикла положить $u(l) := 0$ и перейти к 2°.

5° Заполнение $Q$ по $v$:

5 *a*. $j := 0$; $l := i$;

5 *b*. $j := j + 1$;

5 *c*. $l := v(l)$; $Q(j) := l$; если $l = i$, то перейти к 5*d*, иначе — к 5*b*.

5 *d*. Выдача $Q$, конец (в $Q$ номера вершин стоят против хода контура).

6°. Контура с вершиной $i$ не существует, конец.

Алгоритм 3 фактически строит ордерево с корнем в вершине $i$: номер компоненты $v$ задает номер следующей вершины, а содержимое компоненты — номер непосредственно предшествующей вершины. Управляющий массив $u$ обеспечивает построение именно дерева: если вершина следует за несколькими, то в дерево включается только одна дуга. Работа прекращается, когда контур замыкается на вершине $i$.

Алгоритм 3 содержит цикл до $m$ операций на шаге 2° и цикл в цикле максимум до $m$ операций на шаге 4°, всего $\sim m^2$ операций.

На рис. 2 приведены орграфы, ассоциированные с матрицами $A_1$ и $A_2$. Применяя алгоритм 3 к $\Gamma(A_1)$ при $i=1$, получаем контур 1—3—6—1; применяя алгоритм к $\Gamma(A_2)$ при $i=1$, получаем контур 1—3—5—8—4—6—1.

Теперь для получения алгоритма 2 надо лишь «окаймить» алгоритм 3 и добавить обработку вектора $x$.

### Алгоритм 2

1°. $x:=0$; $M:=1, 2, \ldots, m$.

2°. Если $M=\varnothing$, то перейти к 10°. $i:=M(1)$.

3°—7°. Шаги 1°—5° алгоритма 3, но вместо «перейти к 6°» на шаге 2° — «перейти к 9°».

8°. Добавить единицу к $x_l$ для $l \in Q$; $M:=M \setminus Q$; перейти к 2°.

9°. Конец. Контура с вершиной $i$ не существует, т. е. уравнение (8) не имеет положительных решений.

10°. Конец. Найдено положительное решение $x$.

Алгоритм 3, примененный к матрице $A_1$, выдает контуры 1—3—6—1; 2—4—7—2, но не находит контура, включающего вершину 5. Формула (8) с матрицей $A_1$ положительного решения не имеет.

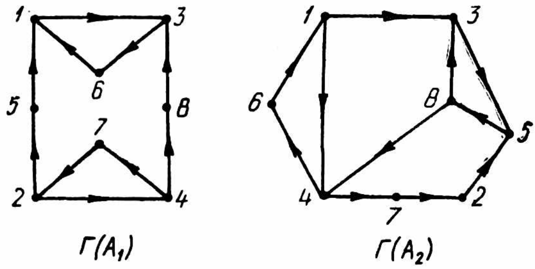


Рис. 2. Орграфы, ассоциированные с матрицами

Алгоритм 3, применяемый к матрице $A_2$, выдает контуры 1—3—5—8—4—6—1, 2—5—8—4—7—2. Уравнение (8) имеет решение $x=(2,\ 1,\ 1,\ 1,\ 2,\ 1,\ 1,\ 1)$. Используя для нормировки равенство (6.1), получаем положительное решение уравнения (4):

$$x=\left(\frac{2}{3},\ \frac{1}{3},\ \frac{1}{3},\ \frac{1}{3},\ \frac{2}{3},\ \frac{1}{3},\ \frac{1}{3},\ \frac{1}{3}\right).$$


**Список литературы:** 1. *Макги Р. Б., Чу-чень-сан.* О выборе типа движения для шагающей системы.— В кн.: Управление в пространстве, М., 1976, с. 187—195. 2. *Mc Ghee R. B., Jain A. K.* Some properties of regularly realizable gait matrices.— Mathematical Biosci, 1972, **13**, № 1/2, p. 179—193. 3. *Охоцимский Д. Е., Голубев Ю. Ф.* Механика и управление движением автоматического шагающего аппарата. — М.: Наука, 1984.—312 с. 4. *Гейл Д.* Теория линейных экономических моделей.— М.: Изд-во иностр. лит., 1963.—418 с. 5. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы / Пер. с англ. — М.: Мир, 1980.—476 с.



*Поступила в редколлегию 18.03.85.*