



ченной системой нелинейных неравенств // Труды междунар. симпозиума «Надежность и качество». Пенза: ПГУ, 2005. С.111-113.

Статья представлена к публикации членом редколлегии О.В. Абрамовым.

УДК 681.5.015.63-192

© 2005 г. **Я.В. Катуева**, канд. техн. наук,
Д.А. Назаров

(Институт автоматизации и процессов управления ДВО РАН, Владивосток)

АЛГОРИТМЫ АНАЛИЗА ОБЛАСТИ РАБОТОСПОСОБНОСТИ, ЗАДАННОЙ В МАТРИЧНОЙ ФОРМЕ¹

Рассматривается задача представления и анализа области работоспособности аналоговых технических устройств и систем, возникающая в процессе проектирования и управления, с учетом параметрических возмущений. Для ее решения предложены алгоритмы построения двух- и трехмерных сечений области (визуализации решения), алгоритм проверки области на дискретность.

Введение

Основные трудности при построении областей работоспособности связаны с большой размерностью пространства варьируемых параметров, следствием которой являются вычислительная трудоемкость соответствующих алгоритмов и сложность интерпретации результатов. В связи с этим практически все известные методы построения областей работоспособности могут реально использоваться при числе варьируемых параметров не больше трех. Кроме того, для многих из них существенным ограничением является отсутствие априорной информации о форме и ориентации области в пространстве параметров, ее выпуклости и односвязности.

Постановка задачи

Задача оптимального параметрического синтеза [1] состоит в выборе номинальных значений внутренних параметров исследуемого устройства $\mathbf{x}_{ном}=(x_{1 ном} \dots, x_{n ном})$, обеспечивающих максимум вероятности его безот-

¹ Работа выполнена в рамках проектов, получивших грант конкурса «Конкурс проектов ДВО РАН 2005 год» по программе № 16 ОЭММПУ и по программе №17 фундаментальных исследований Президиума РАН «Параллельные вычисления и многопроцессорные вычислительные системы».

казной работы в течение заданного времени эксплуатации:

$$x_{ном} = \arg \max P\{\mathbf{X}(x_{ном}, t) \in D_{\mathbf{x}}, \forall t \in [0, T]\}. \quad (1)$$

В (1) $\mathbf{X}(x_{ном}, t)$ – случайный процесс изменения параметров; $D_{\mathbf{x}}$ – область работоспособности; T – заданное время эксплуатации устройства. Область работоспособности $D_{\mathbf{x}}$, как правило, неизвестна, поэтому условия работоспособности обычно задаются системой неравенств:

$$a_j \leq y_j(\mathbf{x}) \leq b_j, \quad j = \overline{1, \dots, m}, \quad (2)$$

где $\mathbf{y} = \{y_j\}_{j=1}^m$ – вектор выходных параметров устройства, причем $y_j = F_j(x_1, \dots, x_n)$, а $F_j(\bullet)$ – известный оператор, зависящий от топологии исследуемого устройства.

За количественный показатель надежности принимается

$$P(\mathbf{Y}(t) \in D_{\mathbf{y}}, \forall t \in [0, T]) = P(\mathbf{y}(\mathbf{X}(t)) \in D_{\mathbf{y}}, \forall t \in [0, T]) =$$

$$= P(\bigcap_{j=1}^m y_j(\mathbf{X}(t)) \in [a_j, b_j] \forall t \in [0, T]).$$

Задача построения области работоспособности может быть сформулирована следующим образом.

Пусть известна система, свойства (качество работы) которой зависят от значений параметров ее элементов $\mathbf{x} = (x_1, \dots, x_n)$, и заданы условия ее работоспособности (2).

Необходимо определить такую область $D_{\mathbf{x}}$ в пространстве внутренних параметров $\mathbf{x} \in R^n$, в каждой точке которой выполняются условия (2).

В работах [2-3] предлагается аппроксимация области работоспособности описанным бруском и представление ее в матричной форме.

Пусть информация об области работоспособности представлена в матричном виде. Описанный брус разбит на множество непересекающихся брусков-ячеек

$$B_0 = \bigcup_{k_1=1}^{l_1} \bigcup_{k_2=1}^{l_2} \dots \bigcup_{k_n=1}^{l_n} B_{k_1, k_2, \dots, k_n},$$

информация о каждом элементе такого многомерного разбиения хранится в массиве $A[R]$, состоящем из нулей и единиц, и векторах-ключах к данной матрице a^0, b^0, l_i , где значения каждого параметра соответственно ограничены минимальными и максимальными значениями a_i^0 и b_i^0 , l_i – число разбиений по каждому их параметров. Шаг по каждому из параметров (длина бруска-ячейки по i -му направлению) составляет

$$h_i = (b_i^0 - a_i^0) / l_i, \quad i = \overline{1, n}.$$

Для дальнейшего решения задачи параметрического синтеза необходимо уметь визуализировать область работоспособности или ее сечения и делать выводы об ее форме (проверять область на непрерывность).

Построение двух- и трехмерных сечений области работоспособности

При количестве входных параметров $n \geq 3$ визуализировать и особенно интерпретировать построенную область работоспособности затруднительно. Однако можно построить двух- и трехмерные сечения данной области, которые могут помочь при решении таких задач.

Как было отмечено, описанный брус B_0 представлен множеством непересекающихся бруско-ячеек. Каждый i -й параметр внутри этого бруса разбит на l_i непересекающихся отрезков-квантов [3].

Сечения любой размерности (меньше размерности матрицы) n -мерной матрицы строятся фиксированием одного или нескольких $k = 1 \dots (n-1)$ параметров на требуемых значениях индексов квантов $q_i = 0 \dots (l_i - 1)$. Остальные $n - k$ параметров считаются свободными для варьирования по всем индексам своих квантов.

Сечением размерности s по параметрам r_1, r_2, \dots, r_k , где $k = n - s$ будем считать множество ячеек матрицы, индексы которых заданы k -фиксированными значениями r_1, r_2, \dots, r_k и s свободными параметрами, каждый из которых принимает все значения индексов своих квантов $q_i = 0 \dots (l_i - 1)$.

Например, при размерности входных параметров $n=4$ необходимо построить двухмерное сечение $s=2$ по параметрам x_1, x_3 ($k=2$). При этом пусть по условию параметр x_1 разбит на 10 квантов внутри описанного бруса B_0 , параметр x_3 разбит на 20 квантов ($l_1 = 10, l_3 = 20$). Имеем два фиксированных параметра, по которым осуществляем сечение области. Сечение производим по первому кванту первого фиксированного параметра и по десятому кванту второго фиксированного параметра, т.е. имеем $r_1=0, r_2=9$ – фиксированные индексы квантов параметров x_1, x_3 . Множество квантов, образованное всеми сочетаниями индексов квантов свободных параметров x_2, x_4 при неизменных значениях индексов квантов фиксированных параметров, представляет собой искомое сечение. Так, если параметр x_2 разбит на 10 квантов внутри описанного бруса B_0 , параметр x_4 разбит на 40 квантов ($l_2 = 10, l_4 = 40$), то сечение будет представлено двухмерной сеткой размером 10×40 ячеек.

Остановимся подробнее на описании алгоритма построения двух- и трехмерных сечений области. Исходные данные: размерность пространства входных параметров системы n , для заданной области построен описанный брус B_0 , каждый i -й параметр внутри этого бруса разбит на l_i квантов, область работоспособности задана ее матричным представлением в виде одномерного массива \mathbf{M} [3]. Алгоритм построения трехмерных сечений приводится схематично для отображения сути их построения. В нем не описываются методы, непосредственно связанные с отображением сечений, в частности, с выбором масштабов для отображения ячеек и пр. В этом

случае фиксированными будут $n-3$ параметров с указанием значений индексов квантов для каждого из них. Имеем массив F_r индексов квантов фиксированных параметров r_1, r_2, \dots, r_k , где $k=n-3$. Также необходим массив P_f с индексами (порядковыми номерами) свободных параметров. Если строим трехмерные сечения, то длина этого массива равна 3. Пробная ячейка матрицы задается массивом $C\{\dots\}$ длины n , в котором индекс элемента соответствует индексу параметра, а значение, находящееся по заданному индексу, содержит номер кванта соответствующего параметра. Например, для четырехмерного случая $C=\{0,0,0,0\}$ означает ячейку матрицы, включающую первые кванты параметров. Перед перебором всех сочетаний индексов квантов свободных параметров запишем в массив C значения индексов фиксированных квантов.

Алгоритм 1. Построение трехмерного сечения n -мерной фигуры:

01. $r = 0$;
02. Цикл i по индексам параметров (от 0 до $n-1$)
03. Если $i \notin P_f$, то
04. $C[i] = F_r[r]$;
05. $r = r + 1$;
06. Конец цикла i ;
07. Цикл i по всем квантам параметра $P_f[0]$.
08. Цикл j по всем квантам параметра $P_f[1]$.
09. Цикл m по всем квантам параметра $P_f[2]$.
10. $C[P_f[0]] = i$;
11. $C[P_f[1]] = j$;
12. $C[P_f[2]] = m$;
13. Если $M[C]$ «хорошая», то
14. Нарисовать куб ячейки C цветом 1;
15. Иначе
16. Нарисовать куб ячейки C цветом 2;
17. Конец цикла m ;
18. Конец цикла j ;
19. Конец цикла i ;

Алгоритм построения двумерных сечений аналогичен, за исключением того, что длина массива P_f будет равна 2, длина массива фиксированных квантов будет равна $n-2$, количество вложенных циклов при переборе сочетаний индексов свободных параметров – на один меньше.

Алгоритм 2. Построение двумерного сечения n -мерной фигуры:

01. $r = 0$;
02. Цикл i по индексам параметров (от 0 до $n-1$)
03. Если $i \notin P_f$, то
04. $C[i] = F_r[r]$;
05. $r = r + 1$;
06. Конец цикла i ;
07. Цикл i по всем квантам параметра $P_f[0]$.
08. Цикл j по всем квантам параметра $P_f[1]$.
11. $C[P_f[0]] = i$;

12. $C[P_f[1]] = j;$
13. Если $M[C]$ «хорошая», то
14. Нарисовать квадрат ячейки C цветом 1;
15. Иначе
16. Нарисовать квадрат ячейки C цветом 2;
18. Конец цикла $j;$
19. Конец цикла $i;$

Построение и визуализация сечений матрицы позволяют облегчить проведение анализа области по ее матричному представлению и сделать предварительные выводы о выпуклости, односвязности области. Предложенные методы и алгоритмы построения и визуализации трехмерного сечения области в 4-мерном пространстве для разных шагов разбиения и двухмерного сечения той же области иллюстрируются рис. 1 и 2.

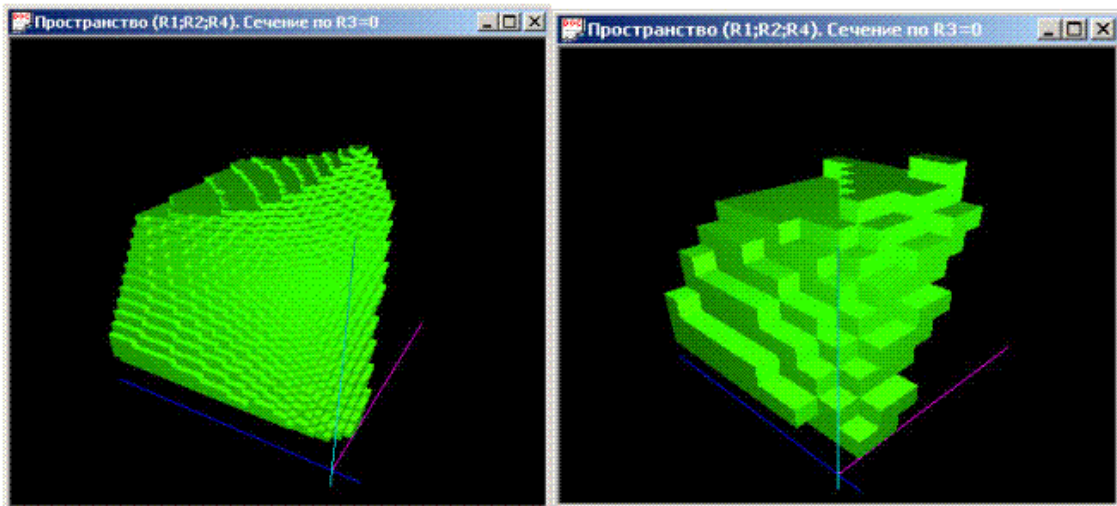


Рис. 1. Трехмерные сечения области работоспособности в 4-мерном пространстве внутренних параметров.

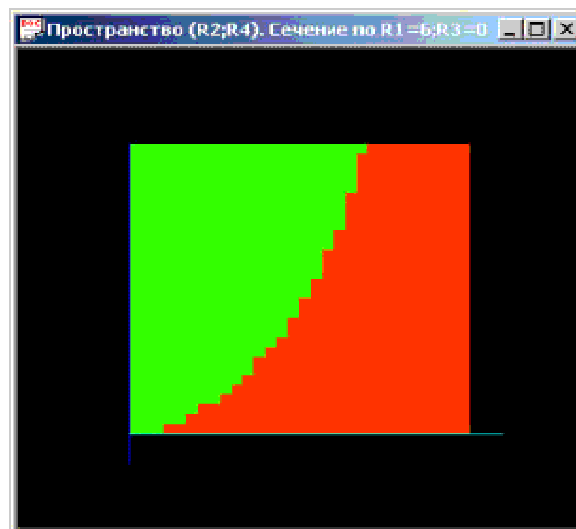


Рис. 2. Двухмерное сечение области работоспособности в 4-мерном пространстве внутренних параметров.

Алгоритм проверки области на неразрывность

Пусть имеется матрица, аппроксимирующая искомую область работоспособности D_x . При числе входных параметров больше трех визуализировать построенную область работоспособности затруднительно. Необходим алгоритм, который бы анализировал имеющуюся матричную аппроксимацию на наличие дискретных областей одного типа (например, анализ множества «хороших» или «плохих» ячеек матрицы [3]). Как результаты этого исследования, так и алгоритм можно использовать для исследования матричного представления области D_x на неразрывность.

Алгоритм обхода однотипных ячеек матрицы заключается в последовательном переборе соседних ячеек одного типа («хорошие» или «плохие»). Ячейкой, соседней данной, будем считать ту, которая располагается слева или справа по какому-либо из координатных направлений от данной ячейки. Ячейка, не лежащая около границы описанного бруса B_0 , имеет в

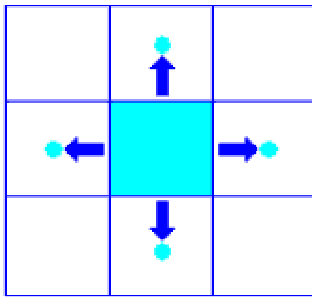


Рис. 3. Выбор соседних ячеек матрицы в двухмерном случае.

двухмерном случае четыре соседние ячейки, в трехмерном – шесть. Для n -мерного пространства количество соседних ячеек равно $2n$.

Если текущая ячейка задана набором индексов $(i_1, i_2, i_3, \dots, i_n)$, то соседние ячейки задаются индексами $(i_1 + 1, i_2, i_3, \dots, i_n)$, $(i_1 - 1, i_2, i_3, \dots, i_n)$, $(i_1, i_2 + 1, i_3, \dots, i_n)$, $(i_1, i_2 - 1, i_3, \dots, i_n)$, $(i_1, i_2, i_3 + 1, \dots, i_n)$, $(i_1, i_2, i_3 - 1, \dots, i_n)$, ... $(i_1, i_2, i_3, \dots, i_n + 1)$, $(i_1, i_2, i_3, \dots, i_n - 1)$. Иллюстрация по выбору соседей текущей ячейки матрицы в двухмерном случае представлена на рис. 3. Если текущая ячейка по одному из индексов является крайней (касается границы описанного бруса B_0), то, естественно, этот индекс не изменяется в направлении границы.

Отметим еще раз, что в матрице «хорошие» ячейки помечаются числом 1, а «плохие» – 0 [3]. При прохождении по однотипным ячейкам необходимо помечать уже пройденные ячейки матрицы. Условимся обозначать числом 3 «хорошие» пройденные и числом 2 – «плохие» пройденные.

Для начала обхода нужно выбрать начальную ячейку матрицы. Поскольку матрица, как описано в [3], хранится в одномерном массиве, поиск начальной точки осуществляется простым перебором элементов этого массива с самого начала, пока не будет найден элемент с требуемым значением, т.е. если, например, необходимо осуществить обход всех «хороших» ячеек, в массиве будет осуществляться поиск значения 1. Выбор направления проводится проверкой всех возможных путей (проверкой всех соседних ячеек). Передвигаться можно только в непройденную ячейку того же типа, что и исходная. Если путей передвижения больше одного, то необходимо запомнить эту ячейку (назовем ее узлом) в массиве узлов.

Массив узлов – это одномерный массив значений индексов матрицы как одномерного массива. В нем хранятся индексы узловых ячеек матрицы. Хранить узловые точки необходимо, чтобы, дойдя до тупиковой ячейки, из которой нет путей, можно было вернуться в последнюю пройденную узловую ячейку для обхода других путей (если представить все пути обхода в виде дерева, то перейти в ближайшую вершину, из которой есть другие пути).

Проверка всех возможных путей перехода от текущей ячейки к следующей выполняется циклическим перебором индексов текущей ячейки матрицы и поочередным уменьшением и увеличением значения индекса.

Алгоритм 3. Выбор направления движения из текущей ячейки:

01. цикл по количеству параметров $k = 1$ до n
02. $i_k = i_k + 1$;
03. проверка возможности перехода к $(i_1, i_2, i_3, \dots, i_k, \dots, i_n)$;
04. если переход возможен, то прекратить цикл.
05. $i_k = i_k - 1$;
06. проверка возможности перехода к $(i_1, i_2, i_3, \dots, i_k, \dots, i_n)$;
07. если переход возможен, то прекратить цикл.

Рассмотрим блок-схему этого алгоритма, приведенную на рис. 4. В ней массив индексов обозначен векторной величиной i . Перед началом алгоритма имеем набор индексов i . Обозначим через M единый индекс матрицы, а массив ее компонентов – **MATR**. Возможность перехода (строки 3 и 6 алгоритма 3) определяется, как уже было описано, по значению ячейки матрицы. Для данного алгоритма непринципиально, «хорошая» она или «плохая», уже пройдена или еще нет. Данный алгоритм ищет ячейку, в которую нужно переместиться, т.е. значения ячеек, которые нужно искать и считать годными для перехода, можно задавать вне этого алгоритма. Обозначим значение искомой ячейки как *val*. После завершения работы алгоритм сигнализирует, найдена ли ячейка для дальнейшего движения или текущая ячейка тупиковая. При этом результирующая переменная *Result* принимает значение TRUE в случае, если следующая ячейка найдена, и FALSE – в противном случае.

Перед переходом к следующей ячейке текущая помечается как пройденная (2, если обходятся «плохие» ячейки, 3 – если «хорошие»). Если из текущей ячейки нет путей дальнейшего перехода (нет соседних непройденных ячеек), то необходимо обратиться к массиву узловых ячеек.

Если массив узловых ячеек пуст, обход закончен. Если он не пуст, нужно взять последний индекс массива матрицы и преобразовать его к набору индексов матрицы. Затем необходимо проверить возможные пути перехода и т.д., как было описано. Если пути дальнейшего движения из текущей ячейки отсутствуют, то нужно удалить последнее значение из массива узловых ячеек. Далее следует переход к последнему элементу массива узловых ячеек.

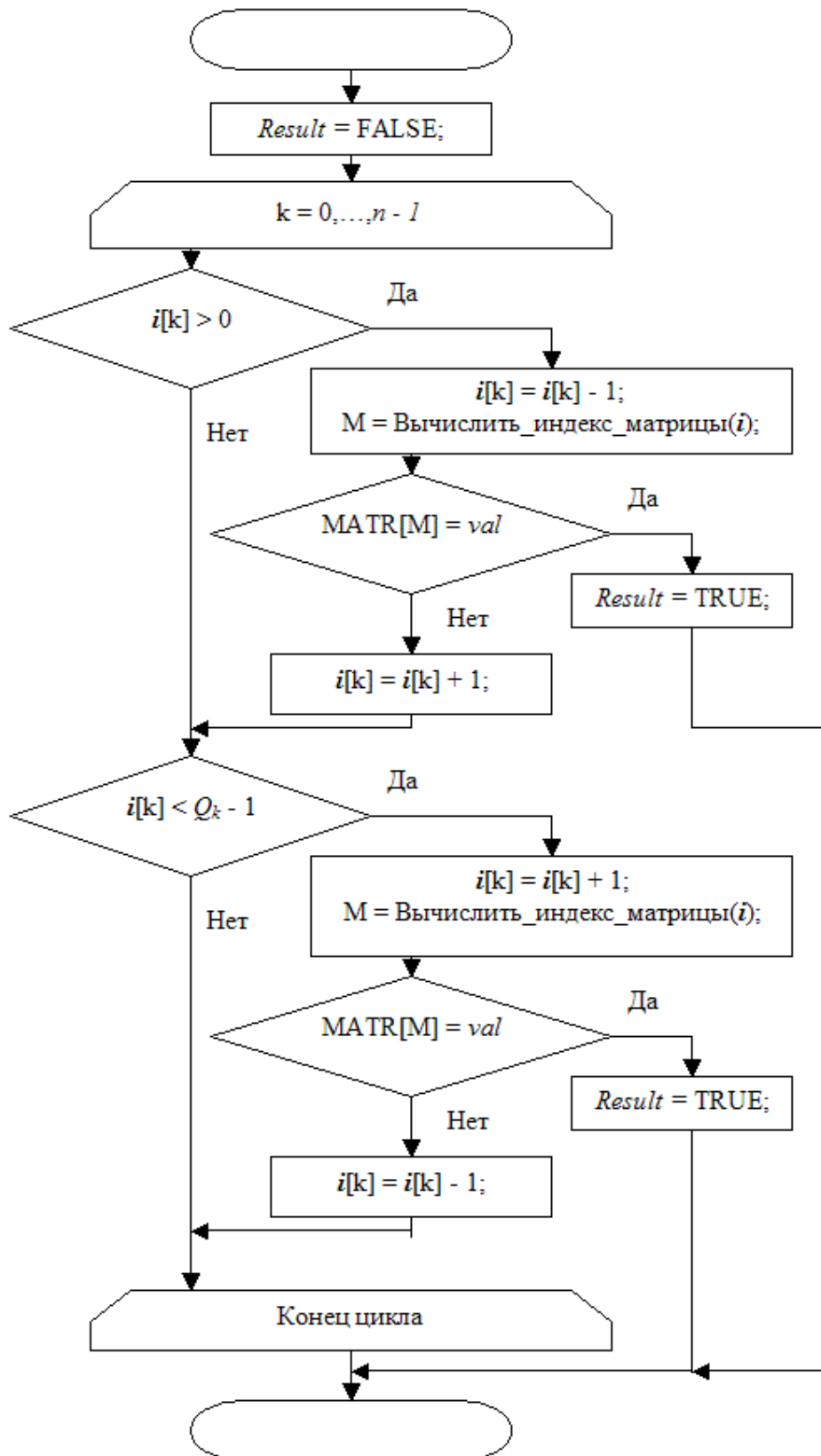


Рис. 4. Блок-схема алгоритма 3 метода выбора направления движения в процедуре обхода однотипных элементов матрицы.

Выше было представлено словесное описание алгоритма. Перед схематичным описанием алгоритма рассмотрим пример обхода в двухмерном случае. На рис. 5 последовательно слева направо показана схема обхода области ячеек, условно помеченных белым цветом.

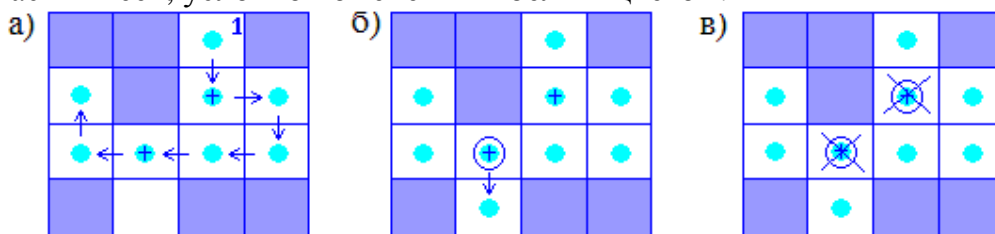


Рис. 5. Схема обхода области в двухмерном случае.

На каждой из трех иллюстраций показан обход до тупиковой ячейки. В приведенном примере ячейки матрицы индексируются стандартным образом: столбцы – слева направо, строки – сверху вниз. Рассмотрим данный пример последовательно и более подробно.

Согласно тому, как в массиве располагаются элементы матрицы [3], в данном случае выбор начальной ячейки очевиден (строки матрицы записываются последовательно, тогда прямым перебором находим первую нужную ячейку). На рис. 5а начальная ячейка помечена цифрой «1».

Выработка направления дальнейшего движения уже была описана выше.

По первому параметру (в данном случае по горизонтали) движение невозможно. По второму параметру (по вертикали) двигаться можно только в сторону увеличения значения его индекса.

Отметили пройденную ячейку. Перешли к следующей.

Из текущей ячейки можно двигаться в двух направлениях, поэтому записываем ее абсолютный индекс в массив узловых ячеек (на иллюстрации узловые ячейки помечены «+»), а также помечаем ее как пройденную ячейку.

Двигаемся по первому возможному направлению. Аналогично продолжаем движение далее.

Встречаем очередную узловую ячейку, отмечаем ее, двигаемся дальше по первому возможному направлению.

Доходим до тупиковой ячейки.

На рис. 5б показан дальнейший ход алгоритма.

В массиве узловых ячеек имеются записи. Берем последний элемент этого массива, обращаемся к ячейке, заданной этим индексом. На рис. 5б эта узловая ячейка обведена окружностью.

Проверяем возможные пути продвижения, находим один.

Передвигаемся в выбранном направлении. Отмечаем ячейку как пройденную.

Движение далее невозможно, ячейка тупиковая.

На рис. 5в показан завершающий этап алгоритма.

Массив узловых ячеек не пуст. Выбираем последний элемент массива.

По выбранному из массива индексу матрицы обращаемся к соответствующей ячейке матрицы.

Поиск дальнейших путей продвижения не дал результатов, поэтому удаляем последний элемент из массива узловых ячеек.

В массиве имеется еще один элемент (после удаления предыдущего он стал последним). Берем последний элемент.

Переходим к соответствующей ячейке матрицы, ищем пути дальнейшего продвижения.

Пути дальнейшего продвижения отсутствуют, поэтому удаляем из массива узловых ячеек последний элемент.

Массив узловых ячеек пуст. Обход области, заданной начальной ячейкой, выполнен.

Рассмотрим приведенный выше алгоритм схематично для пространства параметров произвольной размерности (рис. 6).

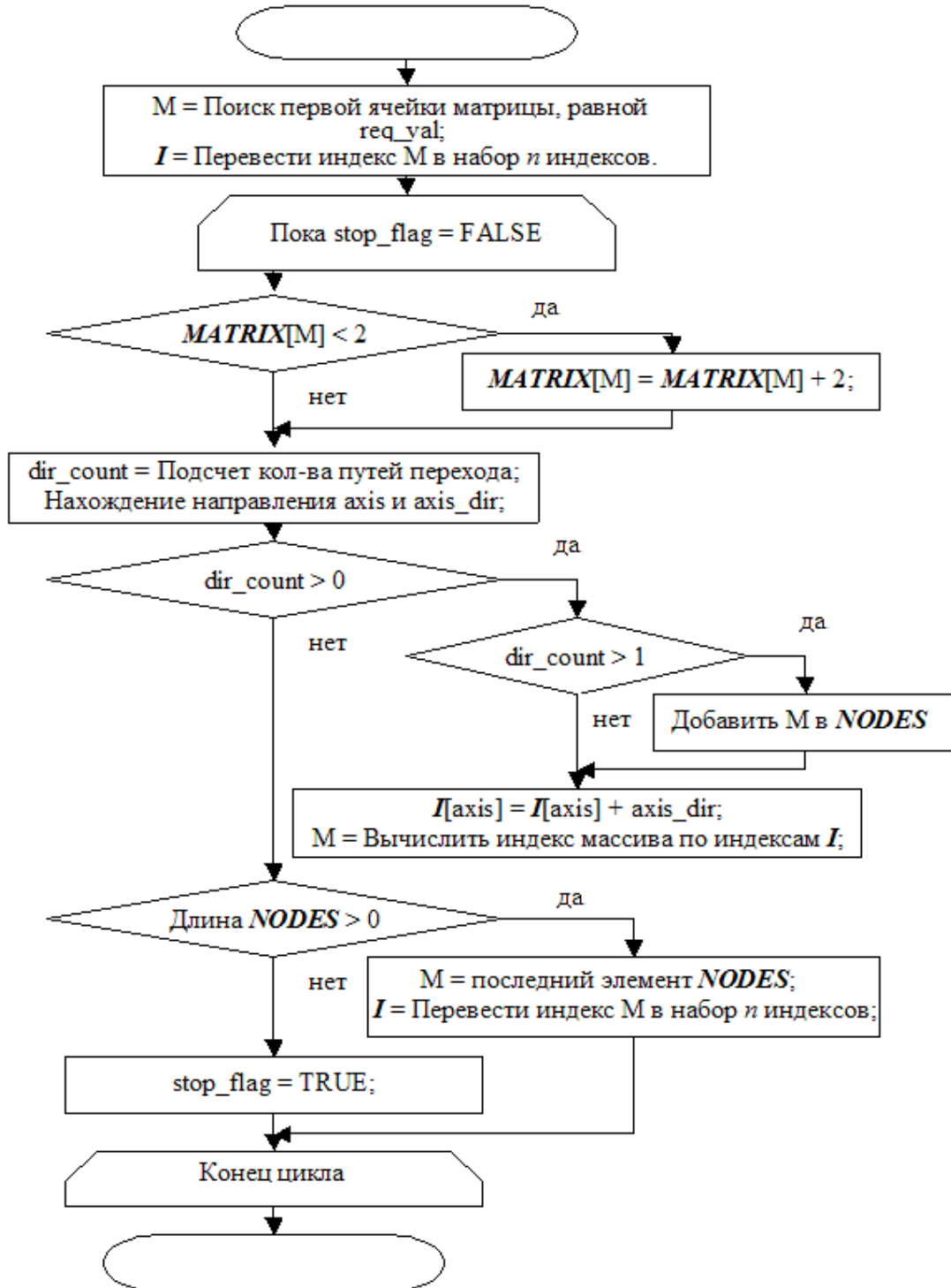


Рис. 6. Блок-схема алгоритма обхода однотипных ячеек матрицы.

Перед этим введем некоторые обозначения: M – индекс массива, хранящего матрицу; I – вектор индексов элемента матрицы в ее n -мерном представлении; $axis$ – индекс координаты, по которой нужно перемещаться к следующей ячейке; $axis_dir$ – направление, в котором нужно перемещаться по оси $axis$ (принимает значения $-1, 1$); dir_count – количество возможных путей перемещения из текущей ячейки; req_val – значения ячеек, по которым нужно делать обход; $NODES$ – вектор (массив) индексов узловых ячеек матрицы; $stop_flag$ – индикатор прекращения цикла обхода; $MATRIX$ – массив, содержащий элементы матрицы. Условимся, что изначально матрица задана ячейками со значениями 0 и 1, где 0 означает «плохую» ячейку, а 1 – «хорошую». Пройденные ячейки будем обозначать соответственно 2 и 3, т.е. 2 – «плохая» пройденная, 3 – «хорошая» пройденная.

ЛИТЕРАТУРА

1. *Абрамов О.В.* Параметрический синтез стохастических систем с учетом требований надежности. М.: Наука, 1992.
2. *Абрамов О.В., Диго Г.Б., Диго Н.Б., Катюева Я.В.* Параллельные алгоритмы построения области работоспособности // Информатика и системы управления. 2004. №2(8). С. 121-133.
3. *Катюева Я.В., Назаров Д.А.* Аппроксимация и построение областей работоспособности в задаче параметрического синтеза // Труды междунар. симпозиума «Надежность и качество». Пенза: ПГУ, 2005. С.130–134.

Статья представлена к публикации членом редколлегии О.В. Абрамовым.