

Рис. 1. Схематический чертеж пространственной засечки определяемого пункта  $K_5$  относительно исходных точек  $K_1 - K_4$

**3. Числовые примеры.** Рассмотрим решение, уравнивание и оценку точности двух пространственных засечек:

1) Пространственная линейная засечка, показанная на рисунке, на котором в скобках даны координаты  $X, Y, Z$  в метрах.

Результаты измерений:  $S_{15} = 60,440$  м;  $S_{25} = 54,332$  м;  $S_{35} = 104,657$  м;  $S_{45} = 38,432$  м.

Методом релаксации из 20 приближений получено:

$$\hat{x} = 24,995 \text{ м}; \hat{y} = 24,995 \text{ м}; \hat{z} = 100,001 \text{ м}.$$

Точностные характеристики при  $n = 2$ ;  $\mu = 0,00538$ :

- полуось эллипсоида ошибок по  $X$ :  $a_x = 0,00235$  м;
- полуось эллипсоида ошибок по  $Y$ :  $a_y = 0,00195$  м;
- полуось эллипсоида ошибок по  $Z$ :  $a_z = 0,00107$  м.

При  $n = 1$  (МНМ) найдено 16 приближений:

$$\hat{x} = 24,997 \text{ м}; \hat{y} = 24,992 \text{ м}; \hat{z} = 99,998 \text{ м}.$$

Точностные характеристики следующие:

- $\mu = 0,00722$ ;
- $a_x = 0,00308$  м;  $a_y = 0,00461$  м;  $a_z = 0,00128$  м.

2) Пространственная засечка по четырем углам наклона, показанная на рисунке.

Результаты измерений:  $v_{15} = 54^\circ 09' 52''$ ;

$$v_{25} = 47^\circ 24' 38''; v_{35} = 71^\circ 04' 39''; v_{45} = 22^\circ 58' 28''.$$

Методом релаксации при  $n = 2$  при  $P = 1$ , из 19 приближений получено:

$$\hat{x} = 24,999 \text{ м}; \hat{y} = 25,001 \text{ м}; \hat{z} = 100,000 \text{ м}.$$

Точностные характеристики следующие:

- $\mu = 5,84$ ;
- $a_x = 0,00188$  м;  $a_y = 0,00183$  м;  $a_z = 0,00076$  м.

При  $n = 1$  (МНМ) найдено 15 приближений:

$$\hat{x} = 24,999 \text{ м}; \hat{y} = 25,002 \text{ м}; \hat{z} = 100,000 \text{ м}.$$

Точностные характеристики следующие:

- $\mu = 6,44$ ;
- $a_x = 0,00332$  м;  $a_y = 0,00154$  м;  $a_z = 0,00141$  м.

**Заключение.** По результатам выполненных исследований можно сделать следующие выводы:

1. Методом релаксации можно одновременно выполнять решение и уравнивание пространственных засечек, заменяя весовые коэффициенты  $C_i$  на  $(P_n)_i$ ;
2. Вертикальная засечка по углам наклона по точности превосходит линейную;
3. В методе наименьших модулей ( $n = 1$ ) результаты оценки точности, как правило, хуже для МНМ, чем для МНК;
4. Актуальной является задача поиска степени  $n_j$  для каждого измерения под условием минимума погрешности положения точки в пространстве  $M = \sqrt{a_x^2 + a_y^2 + a_z^2}$ ;
5. Нелинейные методы устойчивы при обработке пространственных засечек плохого качества. Например, если точки  $K_1 - K_4$  находятся в одной горизонтальной плоскости, то по замкнутым формулам произойдет деление на 0.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Мицкевич, В. И. Вычисление различных видов засечек на ЭЦВМ методом сверхрелаксации / В. И. Мицкевич // Геодезия и картография. – 1974. – № 10. – С. 36 – 40.
2. Мицкевич, В. И. Общий алгоритм вычисления пространственных засечек на ЭВМ методом релаксации / В. И. Мицкевич // Геодезия и картография. – 1978. – № 2. – С. 25 – 28.
3. Абу Дака Имад. Оценка точности пространственных засечек методами нелинейного программирования / Абу Дака Имад, В. И. Мицкевич // Геодезия и картография. – 1994. – № 1. – С. 22 – 24.
4. Мицкевич, В. И. Математическая обработка геодезических сетей методами нелинейного программирования / В. И. Мицкевич. – Новополоцк: ПГУ, 1997. – 64 с.

Материал поступил в редакцию 17.12.07

GRISCHENKOV E.V., ZUEVA L.F., SINJAKINA N.V. Solution, equating and estimating of dimensional geodesic cuts' accuracy using Lp-estimation method

The authors analyze the ground-based dimensional geodesic cuts using Lp-estimation method in view of two range of accuracy:  $n=2$  (Least squares method);  $n=1$  (Least modules method). The decision on the method was made taking into account the lack of redundant measurements. The inaccuracy distribution law can't be found for the abovementioned measurements. The investigation results indicated that nonlinear methods are the most unaffected while determining of awkward shape cuts' position.

УДК 372.800.26.046.14

**Шуть В.Н.**

## ГЕНЕРАЦИЯ ДЕРЕВЬЕВ

**Введение.** Деревья образуют особый класс ациклических графов с жёстким соотношением числа вершин  $n$  и рёбер  $k$ , определяемых соотношением  $k = n - 1$ .

Деревья в программировании используются значительно чаще,

чем другие графы. Так, на построении деревьев основаны многие алгоритмы сортировки и поиска. Компиляторы в процессе перевода программы с языка высокого уровня на машинный язык представляют фрагменты программы в виде деревьев, которые называются

**Шуть Василий Николаевич**, кандидат технических наук, доцент кафедры интеллектуальных информационных технологий Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

синтаксическими. Деревья естественно применять всюду, где имеются какие-либо иерархические структуры, т.е. структуры, которые могут вкладываться друг в друга.

Известно немало задач в физике, химии, биологии, экономике, статистике и лингвистике, сводящихся к подсчёту числа графических объектов, обладающих заданными свойствами. Такой класс задач определён в теории графов как перечисление графов. Некоторые из таких задач были решены около ста лет назад, другие не решены до настоящего времени [1]. Найти в явной форме число соответствующих объектов, как правило, не удаётся. Поэтому программная реализация (генерация) таких объектов и последующий их подсчёт являются единственным методом их перечисления. Отсюда актуальной остаётся разработка «экономических алгоритмов» по числу операций и объёмам используемой памяти.

**Постановка задачи.** Каждому дереву можно поставить в соответствие некоторый код. С помощью этого кода можно восстановить дерево с точностью до изоморфизма. Существуют различные способы кодировки деревьев, которые позволяют решать конкретные задачи (подсчёт деревьев, установление изоморфизма, генерирование всех неизоморфных деревьев и т.д.). Представлением дерева называется способ записи информации о нем, однозначно и полностью восстанавливающий структуру дерева и позволяющий вычислить его характеристики. Выбор представления зависит от решаемой задачи и способа ее решения. Рассмотрим наиболее распространенные способы задания деревьев.

Представление с помощью матрицы смежности является общим для всех видов графов; оно задает граф с точностью до изоморфизма, но вместе с тем данное представление неэкономично, так как ненулевыми являются для  $n$  - вершинного дерева только

$2 \times n - 2$  из  $n^2$  элементов матрицы.

Для неориентированного графа матрица смежности симметрична относительно главной диагонали, поэтому можно задавать только верхнюю треугольную половину матрицы, но это не улучшает ситуацию. Другой недостаток этого представления заключается в том, что трудоёмкость алгоритмов, работающих с таким представлением, достаточно высока.

В представлении с помощью списков смежности каждой вершине дерева сопоставляется список смежных вершин вида

$v_{i1}, v_{i2}, \dots, v_{in}$ .

Для дерева (рис. 1) списки смежности имеют вид:

$v_4 : v_1, v_2, v_3, v_5;$

$v_5 : v_4, v_6, v_7;$

$v_7 : v_5, v_8, v_9.$

$$\left. \begin{array}{l} v_1 : v_4; \\ v_2 : v_4; \\ v_3 : v_4; \\ v_6 : v_5; \\ v_8 : v_7; \\ v_9 : v_7. \end{array} \right\} \text{— не обязательно}$$

При машинной организации списки смежности могут быть связаны между собой разными способами, например, копируя структуру дерева.

Наиболее эффективным для данной задачи является субординационный метод [2].

Суть метода в том, что для каждой вершины записывается величина её мощности (рис. 2). Затем, начиная с большей, записывается строка, для старшей вершины (именуемой корнем) записываются её потомки (сыновья), начиная с большего брата, по очереди начиная со старшего брата, записываются следующие точки (внуки) и т.д.

Для рис. 2 дерево характеризуется записью 431113111.

Алгоритм построения дерева из кода обратен:

1. Для первой цифры рисуем столько потомков, сколько её значение.

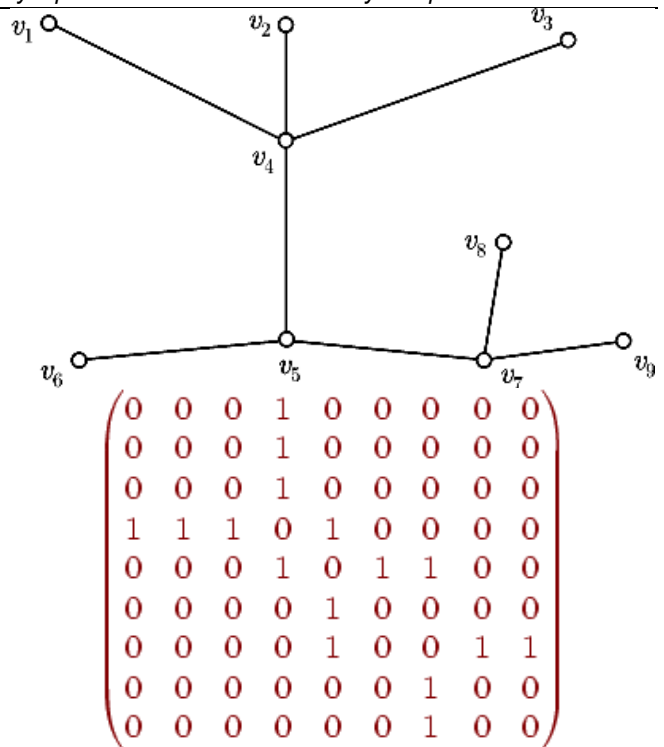


Рис. 1. Пример дерева и его матрица смежности

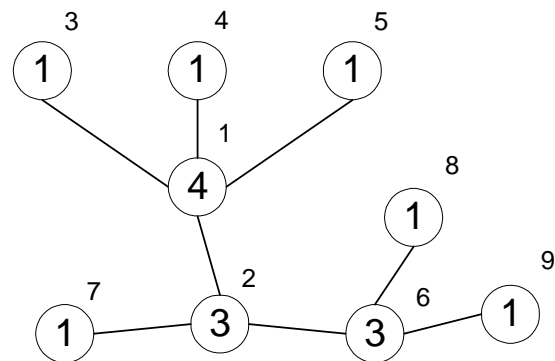


Рис. 2. Пометка вершин дерева их мощностями

2. Для каждого потомка (начиная со старшего) записываем его значение  $n$  и рисуем  $n-1$  его потомков.
3. Повторять п. 2 до тех пор, пока не закончатся цифры кода (рис. 3). Необходимо отметить, что для данного кода построение неизоморфных деревьев невозможно.

**Назначение и функции программы генерации.** Система предназначена для генерации деревьев с любым числом вершин  $n$ . Для этого в систему вводится дерево нулевого уровня, после чего программа генерирует все деревья.

К общим функциям конечной программы будут относиться такие части, как:

1. Генерация потомков для заданного дерева.
2. Сортировка промежуточного результата.
3. Удаление повторяющихся деревьев.
4. Удаление некорректно записанных кодов.
5. Вывод результатов.

Для представления деревьев используется субординационный метод, т.к. он может отображать уникальные деревья и прост в машинной реализации. При генерации деревьев с  $n+1$  вершин из дерева  $n$  вставляется по одному потомку для каждой вершины, кроме тех вершин, у которых:

1. Ветвь симметрична какой-либо ветви и не первична.
2. Мощность вершины равна мощности корня.

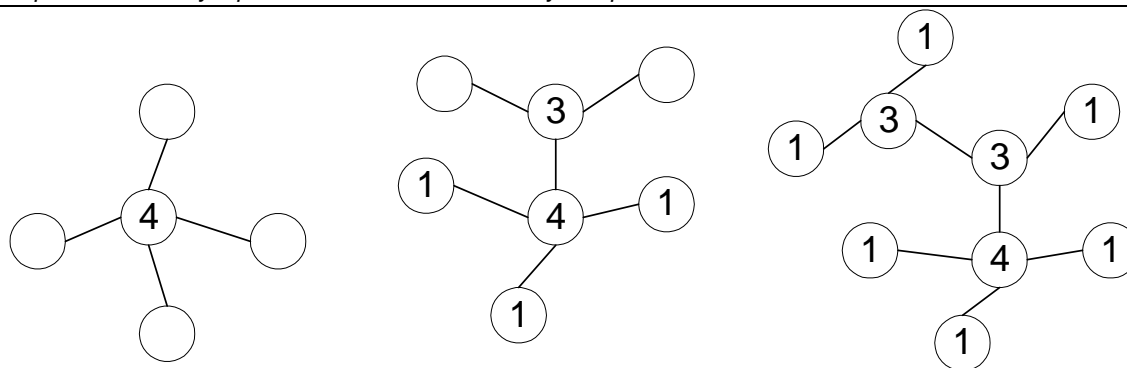


Рис. 3. Построение дерева из субординационного кода

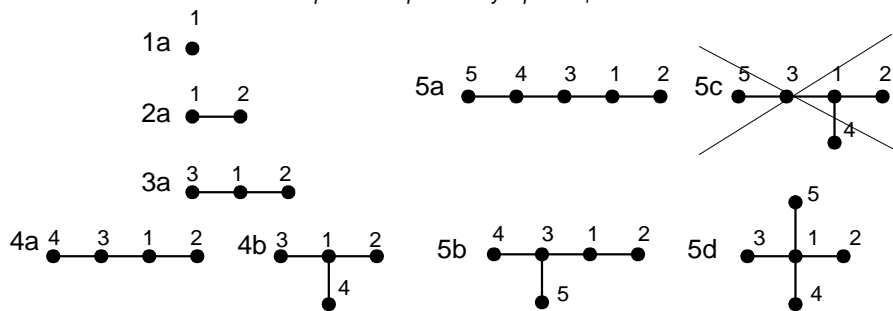


Рис. 4. Пример генерации деревьев

Рассмотрим этот алгоритм для генерации деревьев с  $n=5$  вершинами (рис. 4):

1. У 1а вершина одна и уникальна, добавляем потомка и получаем 2а.
2. У 2а вершина 1 симметрична вершине 2, поэтому добавляем потомка только для первой вершины, получаем 3а.
3. У 3а 1 – уникальна, 2 – симметрично 3, добавляем для 1 и 2, получаем 4а и 4б.
4. 4а: вершины 3 и 4 симметричны 2 и 1 соответственно – добавляем к 3 и 4.  
4б: вершины 3, 4 и 2 симметричны, 1 уникальна – добавляем к 1 и 3.
5. Деревья 5b и 5c изоморфны, одно из них удаляем.

**Проектирование программы.** При реализации данного проекта использована модульная технология проектирования. В основе её лежит разбиение программы на отдельные подпрограммы. Потoki данных между функциями системы изображены на рис. 5.

Опишем глобальные переменные системы:

```
typedef struct { //структура типа дерево
    int node_amo; //кол-во вершин
    int node[mxp]; //вершины
} _tree;
typedef struct { //структура типа узел
    int ancestor; //предок для узла
    int scion[mxp]; //потомки для узла
    int brat; //идентификатор для братьев
    bool fetus; //производность узла
    _tree ntree; //субдерево для узла
} _node;
```

Программа состоит из нескольких подсистем:

```
//воспроизводство всех деревьев child из parent
void reproduct( _tree parent, _tree * const child, int *count );
//сортировка деревьев
void sorting( _tree * const a, int size );
//удаление некорректных кодов
void parent_competition( _tree * child, int *size );
```

**Заключение.** В результате выполнения данной работы был реализован алгоритм генерации деревьев с неограниченным числом вершин  $n$ . Это позволяет программным способом установить полное число неизоморфных деревьев с заданным числом вершин  $n$ . Полученные



Рис. 5. Потoki данных в системе

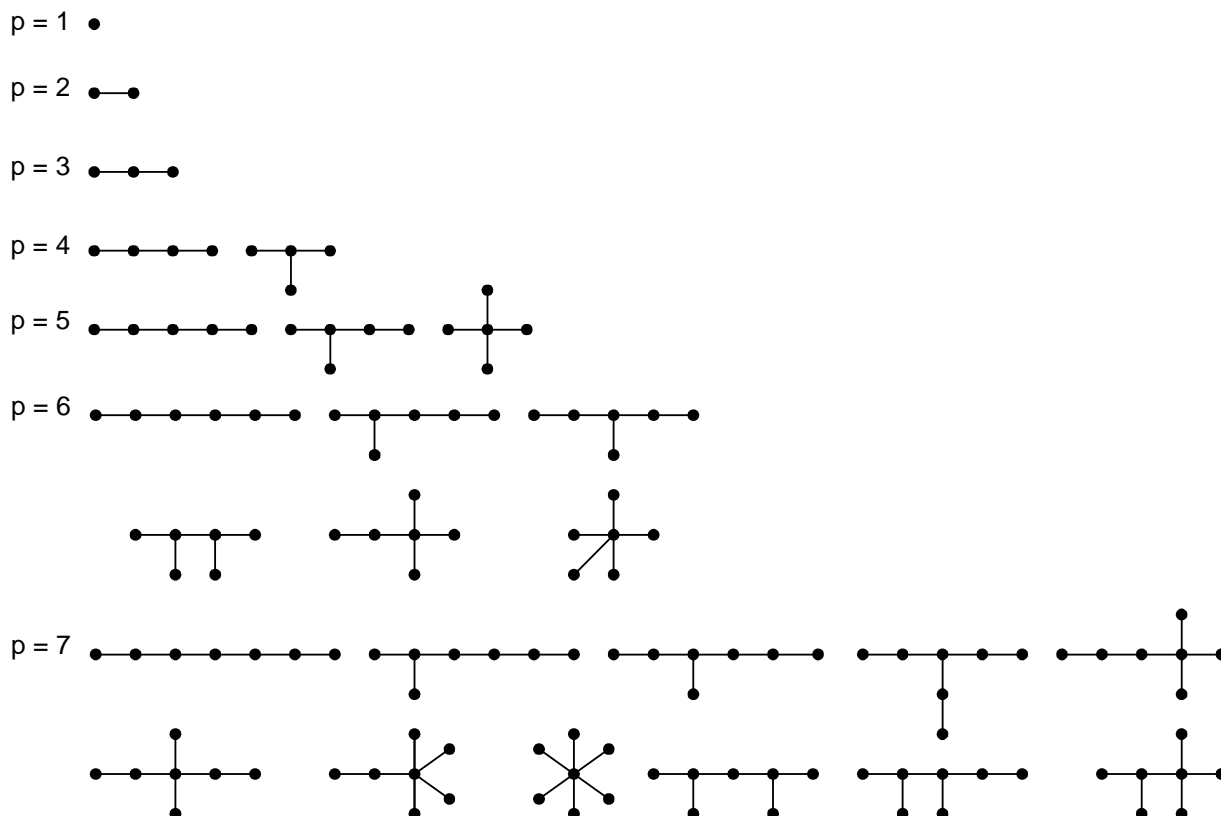
структуры возможных деревьев можно использовать в органической химии, биологии, фармацевтике при построении стереометрических формул сложных молекул. В приложении 1 приведены структуры деревьев для  $n=7$ , полученные данным программным продуктом.

**СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ**

1. Ф. Харари, Э. Палмер. Перечисление графов – М.: Мир, 1977.
2. Статья «Представление деревьев», со страницы <http://www.intuit.ru/departments/algorithms/graphsuse/11/>

Материал поступил в редакцию 28.01.08

## Приложение 1



## SHUT V.N. Generation of trees

In clause the technique of calculation of number of the graphic objects having given properties is considered. Such class of tasks is determined in the theory grafov as transfer grafov. The effective algorithm of transfer grafov of a type a tree is developed.

УДК 681.324

Брюхомицкий Ю.А., Казарин М.Н.

## МНОГОФАКТОРНАЯ СИСТЕМА ПАРОЛЬНО-КЛАВИАТУРНОЙ АУТЕНТИФИКАЦИИ

Персонализация (идентификация) и подтверждение подлинности (аутентификация) субъектов являются одним из основополагающих принципов обеспечения безопасности функционирования автоматизированных информационных систем (АИС).

Процедура идентификации/аутентификации основана на предъявлении субъектом, по крайней мере, одной из трех сущностей:

- нечто, что он знает;
- нечто, чем он владеет;
- нечто, что есть часть его самого.

Первый принцип реализуется в *парольных системах* идентификации/ аутентификации. Эти системы наиболее просты, при условии правильной организации подбора и использования паролей, являются достаточно надежными и потому широко распространены. Основной недостаток парольных систем заключается в принципиальной оторванности аутентификатора от субъекта-носителя. В результате пароль может быть позаимствован тем или иным способом у законного владельца и использован злоумышленником.

Второй принцип реализуется в интеллектуальных замково-ключевых устройствах, которые образовали класс *персональных средств идентификации*. В них, по существу, совмещаются два

тапа идентификатора: парольного и персонального, поэтому их называют еще двухфакторными. Представителями персональных идентификаторов являются токены различных типов и смарт-карты. Двухфакторные средства аутентификации исключают возможность подобрать, подсмотреть или перехватить пароль для входа в АИС, однако персональный идентификатор может быть просто утрачен.

Третьему принципу отвечают только *биометрические параметры организма человека*. При этом для идентификации используются статические параметры: геометрия лица и кистей рук, дактилоскопические узоры пальцев, ладоней и других областей кожи, рисунки радужной оболочки глаза и глазного дна, термограммы артерий и вен, а также динамические параметры: тембр голоса, рукописный и клавиатурный почерки.

Биометрические системы аутентификации имеют ряд неоспоримых преимуществ: биометрические признаки очень трудно фальсифицировать; уникальность биометрических признаков обеспечивает очень высокую достоверность аутентификации; биометрический аутентификатор всегда совмещен с его носителем. Вместе с тем, использование в биометрических системах высоких технологий обуславливают их сравнительно высокую стоимость. Преимущественно это касается систем, использующих статические признаки. Кроме

**Брюхомицкий Юрий Анатольевич**, кандидат технических наук, старший научный сотрудник, доцент кафедры безопасности информационных технологий Таганрогского технологического университета Южного федерального университета.

**Казарин Максим Николаевич**, кандидат технических наук, доцент кафедры безопасности информационных технологий Таганрогского технологического университета Южного федерального университета.

Россия, ТТИ ЮФУ, 347928, г. Таганрог, ул. Чехова, 2.

Физика, математика, информатика