

# ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

УДК 519.16

*А. Н. Белаи*

Северо-Кавказский государственный технический университет  
Ставрополь

## АЛГОРИТМ $\alpha_1$ ПОЛУЧЕНИЯ ОПТИМАЛЬНОГО ИЗОБРАЖЕНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ МОДЕЛИ ПРОГРАММНОГО КОМПЛЕКСА ПО КРИТЕРИЮ СОВПАДАЮЩЕГО НАПРАВЛЕНИЯ РЕБЕР

### Введение

В настоящее время графовое представление информации все больше и больше используется в различных областях точных и естественных наук. В программировании графовые модели применяются при создании программного обеспечения (управляющие графы, иерархии классов, диаграммы потоков данных), дизайне баз данных (диаграммы сущностей-связей), разработке информационных систем и систем реального времени (модели компьютерных сетей, графы состояний, сети Петри), а также в системном программировании – при создании теории компиляции и преобразования программ. Вопрос визуализации подобных графовых структур является краеугольным камнем в процессе адекватного отображения информации.

### Постановка задачи

Пусть объектно-ориентированная модель некоторого программного комплекса  $P$  представлена в виде ациклического графа  $G = (V, E)$ . В общем виде  $|V|$  – множество вершин графа  $G = (V, E)$ ,  $|E|$  – множество ребер графа  $G = (V, E)$ . Но в применении к практической задаче построения объектно-ориентированной модели программного комплекса  $P$ ,  $|V|$  – число классов  $v_i \in V$  программного комплекса  $P$ , а  $|E|$  – число линий связей  $e_i \in E$ , которые характеризуют отношения простого наследования и множественного наследования в комплексе  $P$ . Если класс  $A$  наследует свои свойства от класса  $B$ , то ориентированное ребро  $e_i \in E$  проводится от класса  $B$  по направлению к классу  $A$ . В данной задаче необходимо расположить ациклический граф программного комплекса  $P$  на плоскости. Такое размещение обозначим  $x$ .

Пусть дан ациклический граф  $G = (V, E)$ . Разделим множество  $V$  на элементы  $L_1, L_2, \dots, L_k$ . Пусть  $u, v \in V$ ,  $(u, v) \in E$ , а также  $u \in L_i$  и  $v \in L_j$ , и верно, что  $i > j$ . Тогда элементы  $\{L_x\} = (L_1, L_2, \dots, L_k)$  будем называть уровнями разбиения множества  $V$ . Само такое разбиение  $\{L_x\}$  будем называть поуровневым разбиением множества  $V$ , относящимся к некоторому размещению  $x$ . Множество таких размещений  $x$  назовем множеством допустимых решений  $X = X(G) = \{x\}$  (МДР).

Применительно к задаче об объектно-ориентированном программировании  $L_i$  ( $i = \overline{1, k}$ ,  $L_i \in L_x$ ) есть подмножество множества  $V$  и представляет собой множество классов, находящихся в подмножестве  $L_i$ .

Допустим, что при построении размещения  $x$  на графе  $G = (V, E)$  выделена некоторая вершина  $v' \in V$ , относительно которой производится размещение графа  $G$  на плоскости. Вершина  $v' \in V$  – это один из классов, относящихся к проекту  $P$ . Проведем на плоскости через вершину  $v'$  вертикальную линию  $\sigma$ , которую назовем линией симметрии. Размещение  $x$  графа  $G = (V, E)$  будем строить относительно линии симметрии  $\sigma$ . Вершины  $v_i \in V$ ,  $i = \overline{1, |V|}$ , будем располагать по обе стороны от линии  $\sigma$  согласно уровням  $L_1, L_2, \dots, L_k$ . При этом будем полагать, что размещение  $x$  будет симметричным, если число классов проекта  $P$  будет равным на уровнях  $L_i \in L_x$ ,  $i = \overline{1, k}$  слева и справа от линии  $\sigma$ . Обозначим  $NX_{lj}$ ,  $NX_{pj}$  – число вершин слева и справа от линии симметрии  $\sigma$  на уровне  $L_j$ .

При размещении вершин графа  $G = (V, E)$  необходимо оптимизировать критерий совпадающего направления ребер, отражающих отношения наследования между классами (вершинами

графа). Для этого направление ребер может быть или восходящим, или нисходящим. Предположим, что иерархическая модель связей классов между собой в проекте  $P$  должна быть восходящей. Для этого введем вектор  $Y$  переменных  $y_{kij}$ , которые определяются по следующему правилу:

$$y_{kij} = \begin{cases} 1, & L_{ki} - L_{kj} < 0, \quad L_{ki}, L_{kj} \in L_x, \\ 0, & \text{в противном случае.} \end{cases}$$

$$k = \overline{1, m},$$

где  $m$  – число ребер графа  $G$ , формально выражает число и направление связей, отражающих отношение наследования в проекте  $P$ ;  $L_{ki}$  – номер уровня начала ребра  $e_k$ ,  $e_k \in E$ ,  $L_{ki} \in L_x$ ;  $L_{kj}$  – номер уровня конца ребра  $e_k$ ,  $e_k \in E$ ,  $L_{kj} \in L_x$ .

При представлении объектно-ориентированной модели проекта  $P$  в виде ациклического графа необходимо минимизировать площадь получаемого изображения  $S_x$ . Для этого необходимо минимизировать произведение  $W_x \times H_x$ , где  $W_x$  – ширина получаемого размещения  $x$ .  $W_x$  определяется как максимальная величина из мощностей множеств  $L_1, L_2, \dots, L_k$ :

$$W_x = \max\{|L_1|, |L_2|, \dots, |L_k|\},$$

где  $H_x$  – высота получаемого размещения  $x$ .  $H_x$  определяется по числу уровней в размещении  $x$ ,  $H_x = |L_x|$ .

Для повышения наглядности и простоты просмотра размещения  $x$  необходимо, чтобы оно размещалось равномерно по площади изображения. Очевидно, что для этого ширина получаемого размещения  $W_x$  должна быть равна его высоте  $H_x$ .

При представлении крупного программного комплекса в виде объектно-ориентированной модели на плоскости графом, велика вероятность получения непланарного графа, т. е. графа, который невозможно уложить на плоскости без пересечения линий связей. Поэтому речь идет о минимизации пересечений  $\Pi_x$  линий связей на размещении  $x$ . Под пересечением ребер графа будем понимать случай, когда ребра имеют общую точку не в вершине графа.

При представлении модели проекта  $P$  в виде ациклического графа необходимо учитывать необходимость прямолинейности ребер. Если представить каждое ребро  $e_i \in E$  графа  $G = (V, E)$  в виде уравнения прямой, то очевидно, что все звенья этой прямой должны иметь один угловой коэффициент, или, иначе говоря, число сгибов прямой должно быть минимальным. Рассмотрим ребро  $e_i \in E$ . Пусть это ребро состоит из  $t$  звеньев. Обозначим:  $k_{i,s}$  – угловой коэффициент звена  $s$  ребра  $e_i \in E$ ;  $L_{i,s,q}$ ,  $L_{i,s,r}$  – номера конечного и начального уровней звена  $s$  ребра  $e_i \in E$ ;  $a_{i,s,q}$ ,  $a_{i,s,r}$  – номера конечной и начальной горизонтальной позиции звена  $s$ . Пусть ребро  $e_i \in E$  состоит из  $k$  звеньев. Если ребра прямолинейны, то угловые коэффициенты всех звеньев будут равны между собой.

Введем вектор  $Z$  переменных, компоненты  $z_{i,s,s+1}$  ( $i = \overline{1, m}$ ,  $s = \overline{1, k}$ ,  $m = |E|$ ) которого определяют результат  $F_6(x)$ .

$$z_{i,s,s+1} = \begin{cases} 1, & |k_{i,s+1} - k_{i,s}| > 0, \\ 0, & \text{в противном случае.} \end{cases}$$

Иначе можно сказать, что коэффициент  $z_{i,s,s+1}$  определяет число сгибов ребра  $e_i \in E$ .

Пусть дан ациклический граф  $G = (V, E)$ . Разделим множество  $V$  на элементы  $L_1, L_2, \dots, L_k$ . Пусть  $u, v \in V$ ,  $(u, v) \in E$ , а также  $u \in L_i$  и  $v \in L_j$  и верно, что  $i > j$ . Тогда элементы  $\{L_x\} = (L_1, L_2, \dots, L_k)$  будем называть уровнями разбиения множества  $V$ . Само такое разбиение  $\{L_x\}$  будем называть поуровневым разбиением множества  $V$ , относящимся к некоторому размещению  $x$ . Множество таких размещений  $x$  назовем множеством допустимых решений  $X = X(G) = \{x\}$  (МДР).

Множество всех размещений  $x$  назовем множеством допустимых решений  $X = X(G) = \{x\}$  (МДР).

Качество размещения  $x$  на графе  $G$  задается векторно-целевой функцией (ВЦФ):

$$F(X) = \{(F_1(x), F_2(x), F_3(x), F_4(x), F_5(x), F_6(x)) | x \in X\}, \quad (1)$$

$$F_1(x) = \sum_{j=1}^{|L_x|} |NX_{lj} - NX_{pj}| \rightarrow \min ,$$

где  $F_1(x)$  – симметричное размещение графа  $G$ ;  $NX_{lj}$ ,  $NX_{pj}$  – число вершин слева и справа от линии симметрии  $\sigma$  на уровне  $L_j$ ,  $L_j \in L_x$ .

$$F_2(x) = \forall_{j>i} \sum_{k=1}^m y_{kij} \rightarrow \min ,$$

где  $F_2(x)$  – совпадающее направление ребер.

Для того чтобы оптимизировать этот критерий, необходимо получить строго восходящее изображение графа  $G$ , т. е. необходимо, чтобы для всех ребер графа  $G$  конечная координата каждого ребра была больше, чем начальная. Чтобы составить целевую функцию, введем вектор  $Y$  переменных, компоненты  $y_{kij}$  ( $k = \overline{1, |E|}$ ,  $i, j = \overline{1, L_x}$ ) которого определяют результат  $F_2(x)$ .

$$y_{kij} = \begin{cases} 1, & L_{ki} - L_{kj} < 0, L_{ki}, L_{kj} \in L_x; \\ 0, & \text{в противном случае.} \end{cases}$$

$$k = \overline{1, m},$$

где  $m$  – число ребер графа  $G$ ;  $L_{ki}$  – номер уровня начала ребра  $e_k$ ,  $e_k \in E$ ,  $L_{ki} \in L_x$ ;  $L_{kj}$  – номер уровня конца ребра  $e_k$ ,  $e_k \in E$ ,  $L_{kj} \in L_x$ .

$$F_3(x) = S_x \rightarrow \min,$$

где  $F_3(x)$  – минимизация площади изображения.

$$S_x = W_x H_x,$$

где  $W_x$  – ширина размещения  $x$ ;  $H_x$  – высота размещения  $x$ .

$W_x$  определяется как максимальная величина из мощностей множеств  $L_1, L_2, \dots, L_k$ ,  $W_x = \max\{|L_1|, |L_2|, \dots, |L_k|\}$ .  $H_x$  – высота получаемого размещения  $x$ .  $H_x$  определяется по числу уровней в размещении  $x$ ,  $H_x = |L_x|$ .

Площадь изображения представим себе в виде прямоугольника, площадь которого определяется произведением ширины  $W_x$  размещения  $x$  и его высоты  $H_x$ .

$$F_4(x) = (W_x - H_x) \rightarrow \min,$$

где  $F_4(x)$  – равномерность распределения вершин по площади всего изображения.

Равномерность распределения вершин по площади всего изображения получается только в случае равенства ширины  $W_x$  и высоты  $H_x$  размещения  $x$ .

$$F_5(x) = \Pi_x \rightarrow \min,$$

где  $F_5(x)$  – минимизация количества пересечений ребер.

Под пересечением ребер графа будем понимать случай, когда ребра имеют общую точку не в вершине графа.

$\Pi_x$  – число пересечений на размещении  $x$ .

$$F_6(x) = \sum_{i=1}^m \sum_{s=1}^{t-1} z_{i,s,s+1} \rightarrow \min ,$$

где  $F_6(x)$  – прямолинейность ребер.

$$k_{i,s} = \frac{|L_{i,s,q} - L_{i,s,r}|}{|a_{i,s,q} - a_{i,s,r}|},$$

$$L_{i,s,q}, L_{i,s,r} \in L_X,$$

где  $m$  – число ребер графа  $G$ ;  $t$  – число звеньев в ребре  $e_i \in E$ ;  $k_{i,s}$  – угловой коэффициент звена  $s$  ребра  $e_i \in E$ .

$L_{i,s,q}, L_{i,s,r}$  – номера конечного и начального уровней звена  $s$  ребра  $e_i \in E$ .

Пусть ребро  $e_i \in E$  состоит из  $k$  звеньев. Если ребра прямолинейны, то угловые коэффициенты всех звеньев будут равны между собой.

Введем вектор  $Z$  переменных, компоненты  $z_{i,s,s+1}$  ( $i = \overline{1, m}, s = \overline{1, k}, m = |E|$ ) которого определяют результат  $F_6(x)$ .

$$z_{i,s,s+1} = \begin{cases} 1, & |k_{i,s+1} - k_{i,s}| > 0; \\ 0, & \text{в противном случае.} \end{cases}$$

Иначе можно сказать, что коэффициент  $z_{i,s,s+1}$  определяет число сгибов ребра  $e_i \in E$ .

Решением задачи (1) является  $\tilde{x}$ , где  $\tilde{x}$  – есть множество Парето  $X$  по векторному критерию  $F(X) = \{(F_1(x), F_2(x), F_3(x), F_4(x), F_5(x), F_6(x)) \mid x \in X\}$ .

Будем предполагать, что исходный граф  $G = (V, E)$  является так называемым  $st$ -графом, т. е. в нем имеется только одна начальная вершина, обозначаемая  $s$ , и только одна конечная вершина, обозначаемая  $t$ . Ясно, что любой дэг  $G = (V, E)$  может быть преобразован в  $st$ -граф добавлением фиктивных вершин  $s$  и  $t$ , а также дуг  $(s, p)$  и  $(q, t)$  для всех начальных вершин  $p$  и конечных вершин  $q$  графа  $G$ . Изображение графа легко конструируется на основе его обзорного представления.

Обзорным представлением  $\Gamma$  заданного  $st$ -графа  $G$  называется такое его изображение, в котором каждая вершина  $p$  представлена горизонтальным отрезком  $\Gamma(p)$ , называемым вершинным отрезком, а каждая дуга  $(p, q)$  – вертикальным отрезком  $\Gamma(p, q)$ , называемым реберным отрезком, таким образом, что справедливы следующие три свойства:

- вершинные отрезки не накладываются друг на друга;
- реберные отрезки не накладываются друг на друга;
- реберный отрезок  $\Gamma(p, q)$  имеет нижнюю границу, лежащую на  $\Gamma(p)$ , и верхнюю границу, лежащую на  $\Gamma(q)$ , и не пересекает ни один другой вершинный отрезок [1].

### Анализ существующих методов решения

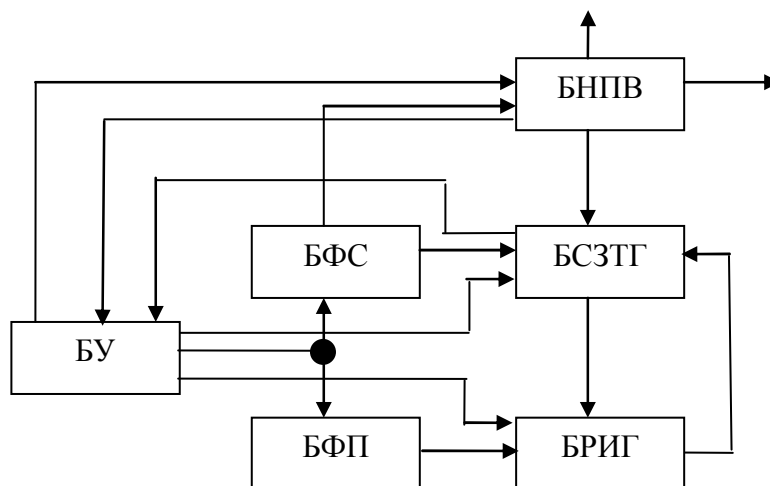
При построении алгоритма, строящего изображение графа на плоскости исследователю приходится решать проблему построения его планарного изображения. К сожалению, определение планарности графа относится к классу  $NP$ -трудных задач [1].

Часто для решения задачи исследования планарности пользуются методом Бадера. При этом предполагается, что в связном графе известен гамильтонов цикл. Задача определения гамильтонова цикла является  $NP$ -трудной.

Другой подход к исследованию планарности графов основан на теореме Понтрягина – Куратовского, которая формулируется так: граф является планарным тогда и только тогда, когда он не содержит подграфа, изоморфного с точностью до вершины степени 2 одному из графов Понтрягина – Куратовского. Эти графы, обозначаемые соответственно  $K_5$  (полный пяти-вершинный) и  $K_{3,3}$ , заведомо непланарны.

Из теоремы Понтрягина – Куратовского следует, что для определения планарности графа из него нужно последовательно выбрать всевозможные сочетания 5- и 6-вершинных подграфов и сравнивать их на изоморфизм с графами  $K_5$  и  $K_{3,3}$  соответственно.

Обобщенная структурная схема модели для определения планарности графов и выделения максимальной планарной части представлена на рисунке.



Аппаратная модель для определения планарности графа:

БНПВ – блок накопления планарных вершин;  
 БСЗТГ – блок селекции и задания топологии графа;  
 БРИГ – блок распознавания изоморфизма графа;  
 БФС – блок формирования сочтаний;  
 БФП – блок формирования перестановок;  
 БУ – блок управления

В общем случае задача определения изоморфизма  $NP$ -трудная, но в данном случае исследованию на изоморфизм подвергаются только 5- и 6-вершинные графы, т. е. графы малой размерности. В этом подходе используется простейший алгоритм распознавания изоморфизма – на основе полного перебора, который требует  $5! = 120$  тактов или  $6! = 720$  тактов при исследовании на изоморфизм 5- и 6-вершинных графов.

Очевидно, что для построения изображения графа с максимальной выделенной планарной частью необходимо использовать приближенный полиномиальный алгоритм. Этот алгоритм отражен в процедуре MAXSUGRAPH алгоритма  $\alpha_1$ .

#### АЛГОРИТМ $\alpha_1$

ВХОД: ациклический граф  $G = (V, E)$ .

ВЫХОД: оптимальное восходящее изображение  $G = (V, E)$ .

1. Процедура MAXSUGRAPH.

Выделить максимальный планарный подграф  $G' = (V', E')$  графа  $G$ ,  $V' \subseteq V$ ,  $E' \subseteq E$ .

2. Для каждой вершины  $v$ ,  $v \in V$  цикл  $c_1$ .

Процедура ПОЛИЛИНЕЙНОЕ ИЗОБРАЖЕНИЕ

Получение изображения  $\beta_i$ ,  $i = 1, |V|$ .

3. Анализ на оптимальность изображения  $\beta_i$ ,  $i = 1, |V|$ .

4. Окончание цикла  $c_1$ .

5. Окончание алгоритма  $\alpha_1$ .

Процедура ПОЛИЛИНЕЙНОЕ ИЗОБРАЖЕНИЕ

ВХОД: ациклический граф  $G = (V, E)$ .

ВЫХОД: изображение  $\beta_i$ ,  $i = 1, |V|$ .

1. Конструируется обзорное представление  $\Gamma$  графа  $G$  с целочисленными координатами вершин.

2. Для каждой вершины  $v$ ,  $v \in V$  цикл  $c_2$ .

3. Заменить вершинный отрезок  $\Gamma(v)$  на произвольную точку  $P(v) = (x(v), y(v))$  отрезка  $\Gamma(v)$ .

4. Окончание цикла  $c_2$ .

5. Для каждого ребра  $(n, v) \in E$  цикл  $c_3$ .
6. Если  $y(v) - y(n) = 1$ , то.
7. % Короткое ребро %. Заменить реберный отрезок  $\Gamma(u, v)$  на отрезок с конечными вершинами  $P(u)$  и  $P(v)$ .
8. Иначе % Длинное ребро %. Заменить реберный отрезок  $\Gamma(u, v)$  на ломаную линию, соединяющую точку  $P(n)$  с точкой  $P(v)$  через точки  $(x(\Gamma(u, v)), y(u) + 1)$  и  $(x(\Gamma(u, v)), y(v) - 1)$ .
9. Окончание цикла  $c_3$ .

Процедура MAXSUGRAPH

ВХОД: ациклический граф  $G = (V, E)$ .

ВЫХОД: максимальный планарный подграф  $G' = (V', E')$  графа  $G$ ,  $V' \subseteq V$ ,  $E' \subseteq E$ .

Создается начальный граф  $G' = (V, \emptyset)$ . Этот граф образуется из графа  $G = (V, E)$  путем представления всех смежных вершин в виде несмежных вершин.

1. Для каждого ребра  $G$  цикл  $c_4$ .
2. Добавляется на каждом шаге по одному ребру из  $G/G'$  без нарушения планарности до тех пор, пока не возникнет ситуация, когда нет ребра в  $G/G'$ , добавление которого к  $G'$  не нарушает свойство планарности  $G'$ .
3. Окончание цикла  $c_4$ .

### Заключение

Таким образом, в работе предложен алгоритм, который решает проблему построения оптимального разбиения ациклического графа по критерию совпадающего направления ребер и получения оптимального изображения объектно-ориентированной модели программного комплекса. Но проблема построения оптимального изображения графа по критерию совпадающего направления ребер в значительной степени сопряжена с проблемой укладки графа на плоскости, которая, в свою очередь, связана с получением минимума пересечений его ребер. Задача выделения максимальной планарной части графа является *NP*-трудной, поэтому был проведен анализ некоторых существующих методов, решающих эту проблему, и в алгоритме  $\alpha_1$  была предложена процедура MAXSUGRAPH, которая решает проблему получения оптимального изображения графа за полиномиальное время  $O(n)$ .

### СПИСОК ЛИТЕРАТУРЫ

1. Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ – Петербург, 2003. – 1104 с.

Получено 16.10.2006

### ALGORITHM OF THE OPTIMUM IMAGE RECEIVING OF AN OBJECT-ORIENTED MODEL OF A PROGRAM COMPLEX DUE TO THE CRITERION OF COINCIDENT DIRECTION OF EDGES

A. N. Belash

Nowadays a graph image of information is getting more and more popular in various fields of exact and natural sciences. In programming graph models are used for software creation (managing graphs, hierarchy of classes, diagrams of data flows), design of data bases (diagrams of essential models-communications), development of information systems and systems of real time (model of computer webs, graphs of conditions, Petri nets), and also in system programming – for the creation of the compilation theory and transformation of programs. Let object-oriented model of a program complex  $P$  be represented as an acyclic graph  $G = (V, E)$ . The quality of placing  $x$  on the graph  $G$  is set by the vector – criterion function:  $F(X) = \{(F_1(x), F_2(x), F_3(x), F_4(x), F_5(x), F_6(x)) | x \in X\}$ .