

УДК 681.3

## О ПРОГРАММИРОВАНИИ ПРИКЛАДНЫХ ЗАДАЧ, СВОДИМЫХ К ЗАДАЧЕ ЦИРКУЛЯЦИИ МИНИМАЛЬНОЙ СТОИМОСТИ

© 2010 г. \*Высоцкая А. А., \*\*Белякова А. А.

\* Южно-Российский государственный технический университет (НПИ)

\*\* ОАО «Глория Джинс», Ростов-на-Дону

*Предложен подход к организации программной реализации задач, сводимых к ЗЦМС и решаемых методом дефекта, основанный на отделении описания потоковой сети от собственно алгоритма дефекта.*

Ключевые слова: задача циркуляции минимальной стоимости; описание потоковой сети; алгоритм дефекта; интерфейс настройки алгоритма дефекта на прикладную задачу; программное обеспечение интерфейса..

*The way of program realization for the tasks, which come to the task of minimal value circulation, and can be solved using the blemish method. This method is based on streaming network description's separating from the blemish algorithm itself.*

Key words: task of minimal value circulation; blemish algorithm; interface of blemish algorithm's tuning for an applied task; software of the interface.

Особое место в теории потоков занимает задача о циркуляции минимальной стоимости (ЗЦМС) и метод ее решения, известный как метод дефекта [1; 2]. Большое число прикладных задач, представленных в работах [1; 3; 4; 9], сводится к ЗЦМС, а, следовательно, алгоритм ее решения имеет достаточно универсальный характер. Особенно выделим практически важный случай, когда задачи, сводимые к ЗЦМС, могут быть собраны в рамках одного проекта (например, система транспортной оптимизации [5], или система планирования размещения продукции в территориально-распределенной сети дистрибуции [6]). При этом ЗЦМС и алгоритм дефекта являются ядром математического и алгоритмического обеспечения. Естественная постановка задачи на разработку подобных систем заключается в том, чтобы одна и та же реализация алгоритма дефекта применялась для решения всех функциональных задач системы.

Сформулируем ЗЦМС в терминах потока. Пусть дана сеть  $G(N, A)$ , где  $N$  – множество вершин,  $A$  – множество дуг, каждая

из которых характеризуется следующими величинами:

$L(i, j)$  – нижняя граница пропускной способности дуги  $(i, j) \in A$ ;

$U(i, j)$  – верхняя граница пропускной способности дуги  $(i, j) \in A$ ;

$c(i, j)$  – стоимость прохождения единицы потока из узла  $i$  в узел  $j$ .

Требуется построить поток  $f$ , удовлетворяющий условиям (1), (2) и минимизирующий линейную функцию стоимости (3).

$$\sum_{j \neq i} f(i, j) - \sum_{j \neq i} f(j, i) = 0 \quad (1)$$

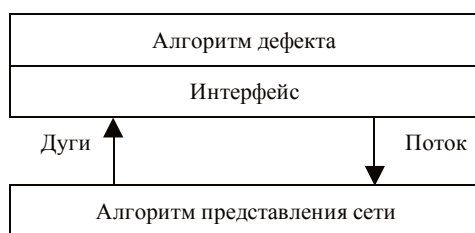
при всех  $i \in N$ ,

$$L(i, j) \leq f(i, j) \leq U(i, j) \text{ при всех } (i, j) \in A, \quad (2)$$

$$\sum_{(i, j) \in A} c(i, j) \cdot f(i, j) \quad (3)$$

Несмотря на несомненную практическую значимость, затрагиваемая проблема в литературе, содержащей описание метода дефекта, практически не рассматривается. Принципиальный момент здесь заключается в том, что описание метода дефекта выполне-

но в терминах ЗЦМС. Поэтому для решения некоторой задачи требуется на абстрактном уровне выполнить ее сведение к ЗЦМС [6], а затем применительно к ЗЦМС реализовать метод дефекта. То есть, в общем случае решение любой задачи требует индивидуальной реализации алгоритма дефекта, программирование которого отличается значительной сложностью и трудоемкостью. Таким образом, проблема эффективного применения метода дефекта актуальна. Именно этой проблеме посвящена статья [5], где рассмотрен особый подход к алгоритму дефекта. Суть его заключается в отделении описания потоковой сети от собственно алгоритма дефекта. Это позволяет с одной стороны обобщить описание сети, сделать более абстрактным, что создает элементы настройки на задачи ЗЦМС, а с другой стороны, сделать более конкретным, ориентированным исключительно на реализацию метода дефекта (рис. 1).



**Рис. 1.** Подход к реализации алгоритма дефекта

Результатом статьи является выделение четко определенного интерфейса который требуется реализовать для настройки алгоритма дефекта на конкретную прикладную задачу.

В развитие подхода предложенного в [5], в данной статье рассмотрены принципы реализации программного обеспечения в популярной системе DELPHI.

При этом в объектно-ориентированной среде DELPHI естественно использовать методы объектно-ориентированного программирования.

Алгоритм дефекта программно реализуется как класс с абстрактными виртуальными методами, реализующими интерфейсные функции (табл. 1).

На этапе решения прикладной задачи методами наследования создается потомок класса с алгоритмом дефекта. Класс-потомок получает все поля и методы класса-родителя, но может дополнять их собственными либо переопределять уже имеющиеся. В порожденном классе перекрываются виртуальные интерфейсные методы. В результате этот порожденный класс и есть программная реализация задачи сводимой к ЗЦМС и решаемой алгоритмом дефекта.

В свою очередь алгоритм дефекта как класс DELPHI должен быть порожден от некоторого базового класса (класс, от которого произошло наследование, называется базовым или родительским). Выбор этого класса не является тривиальным. Отметим, прежде всего, что свойства и методы этого базового класса наследуются не только алгоритмом дефекта, но и порожденным классом, реализующим решаемую задачу.

Решаемая задача в информационном аспекте реализуется в системе клиент-сервер. Система DELPHI имеет развитые средства поддержки технологии клиент-сервер. От-

Таблица 1

### Функции интерфейса I

Функции	Применение
Определение количества вершин в сети	Применяется на начальном этапе для формирования множества пометок и множества узловых чисел
Инициализация просмотра дуг	Применяются в процедурах поиска дефектной дуги, расстановки пометок, корректировки узловых чисел для последовательной генерации всех дуг сети
Выбор очередной дуги и определение ее параметров	
Запоминание дуги	Для выдачи метки текущей дуге
Восстановление дуги	Для восстановления дуги по ее метке

метим, прежде всего, не визуальные компоненты-наследники TDataSet, которые отображают наборы данных, и в частности, SQL-запросы. Именно через такие наборы данных должен быть реализован интерфейс с виртуальными функциями в классе решаемой задачи. Как правило, невизуальные компоненты наборов данных размещают в модуле данных TDATAMODULE, который как раз для этих целей и был включен в DELPHI.

Отсюда следует, что модуль данных TDATAMODULE наиболее подходящий «кандидат на роль» базового класса для алгоритма дефекта.

Однако, TDATAMODULE не просто класс, а контейнер для невизуальных компонентов на этапе разработки и, следовательно, состоит не только из модуля UNIT, но и файла DFM, который отображает размещенные в модуле данных компоненты. Для таких классов в DELPHI штатным средством наследования является визуальное наследование [7] которое для класса потомка дополнительно создает по определенным правилам свой DFM-файл.

Визуальное наследование является мощным средством расширения функциональности существующих классов не только за счет исполняемого кода но и используемых компонент и их свойств.

Однако в рассматриваемой задаче метод визуального наследования является излишне громоздким и неэффективным, поскольку именно визуальное наследование здесь не требуется.

Таким образом, с одной стороны использование TDATAMODULE в качестве базового класса для алгоритма дефекта позволяет сохранить все преимущества визуального проектирования в DELPHI, а с другой стороны требует нестандартного для DELPHI подхода к наследованию.

В качестве такого метода в данной статье предлагается использовать понятие «встраиваемого класса» [7].

Алгоритм дефекта этим методом реализуется в модуле UDMdefect. Подчеркнем что это обычный модуль (unit) без файла DFM.

Реализация метода «встраиваемого класса» заключается в следующем.

1) В модуле UDMdefect объявляем новый класс

```
Type
TDataModule = class (Classes.
TDataModule)
```

В этом классе как раз и реализуется алгоритм дефекта.

2) Для решаемой прикладной задачи создаем обычный модуль данных, в котором размещаем необходимые компоненты для наборов данных.

3) В предложении USES созданного модуля данных делаем ссылку

```
Uses UDMdefect,
```

где UDMdefect – модуль, содержащий определение и реализацию класса алгоритма дефекта.

4) После в модуле данных класс TDataModule наследуется не от стандартного Classes.TDataModule, а от UDMdefect.TDataModule, то есть от класса реализующего алгоритм дефекта.

Декларируем и реализуем виртуальные методы решаемой задачи для алгоритма дефекта.

Для обоснования применения метода встраиваемого класса, прежде всего, отметим, что конкретизация имени через указание UNIT, в котором это имя определено, является документированной возможностью Паскаля с тех самых пор, как в нем было введено разбиение программ на модули.

Выше было использовано еще одно правило.

По правилам языка если в uses есть два модуля, содержащих одно и то же имя, то будет использован элемент из того модуля, который в списке uses указан позже. То есть путем указания в предложении uses ссылки на UDMdefect после Classes достигается требуемый эффект.

В целом предлагаемый метод основан на разделении функций дизайнера и компилятора.

Когда компоненты размещаются в модуле данных и задаются их свойства, дизайнер вставляет в определенные места кода нужные строки и создает DFM-файл с информацией о компонентах и их свойствах, по которому будет создаваться модуль данных. В компиляции дизайнер не участвует, соответственно, компилятор использует только код. А в коде используется другой класс – потомок первоначального – с тем же именем и тем же

набором свойств. В Паскале использование одинаковых имен ошибкой не является, так как существуют формальные правила разрешения таких ситуаций.

Таким образом, в системе DELPHI для решения задачи ЗЦМС необходимо:

1)определить структуры данных задачи в модуле данных;

2)по методике описанной в статье сделать модуль данных от класса TDataModule, в котором реализован алгоритм дефекта;

3)разработать производный класс реализующий решаемую прикладную ЗЦМС, путем реализации интерфейсных функций;

4)разработать интерфейс задачи, включающий в себя вызов алгоритма дефекта и интерпритацию результатов его решения.

Отметим, что последний пункт эффективно решается средствами визуального построения пользовательского интерфейса в системе DELPHI.

#### Литература:

1. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. –М.: Мир, 1984. – 494 с.

2. Форд Л., Фалкерсон Д. Поток в сетях. –М.: Мир, 1966. – 274 с.

3. Йенсен П., Барнес Д. Поток в сетях. Программирование. –М.: Радио и связь, 1984. – 392 с.

4. Белякова А. Декомпозиционный подход к реализации Алгоритма Дефекта // Изв. вузов. Сев.-Кавк. регион. Техн. науки. – 2003. – №2. – С.12-18.

5. Белякова А. А. Об эффективной организации информации в системе транспортной оптимизации // Изв. вузов. Сев.-Кавк. регион. Техн. науки. – 2002. – №2. – С. 7-9.

6. Высоцкая А. А., Белякова А. А. Решение задачи планирования размещения продукции методами потокового программирования // Моделирование. Теория, методы и средства. Материалы IX Междунар. науч.-практ. конф., г. Новочеркасск 13 апр 2009 г. / Юж.-Рос. гос. техн. ун-т (НПИ). –Новочеркасск: ЮРГТУ, 2009. – С. 13-19.

7. Кэнту М. Delphi 7 для профессионалов. –СПб.: Санкт-Петербург, 2004. – 1104 с.

8. Дарахвелидзе П. Г., Марков Е. П. Delphi 4.–СПб.: Санкт-Петербург, 1999.– 816 с.

9. Высоцкая А. А., Канивец П. И. Решение задач размещения продукции методами потокового программирования // Студенческая научная весна – 2009; Материалы межрегиональной научно-технической конференции студентов, аспирантов и молодых ученых Южног федерального округа / Юж.-Рос.гос. техн.ун-т. –Новочеркасск: ЮРГТУ (НПИ), 2009. – С. 406-408.

Поступила в редакцию

17 января 2010 г.



**Анна Александровна Высоцкая** – аспирант кафедры «Прикладная математика» ЮРГТУ (НПИ). Автор работ по проблеме размещения продукции в территориально распределенной сети дистрибуции методами потокового программирования. Научный руководитель – зав.каф. ПМ ЮРГТУ (НПИ), д.т.н., проф. А. Н. Ткачев, научный консультант – докторант ЮРГТУ (НПИ), к.э.н. П. И. Канивец.

**Anna Aleksandrovna Vysotskaya** – postgraduate student of SRSTU (NPI) «Applied mathematics» department. Author's works are dedicated to problem of products' allocating in geografically distributed network, using the methods of streaming programming. Research supervisor – chief of SRSTU (NPI) «Applied mathematics» department, Ph.D., doctor of technics, professor A. N. Tkachyov, research adviser – SRSTU (NPI) competitor for doctor's degree, Ph.D., candidate of economics P. I. Kanivets.

346405, г. Новочеркасск, ул. Свободы, д. 23, кв.72  
23 Svobody st., app. 72, 346405, Novocherkassk, Rostov reg., Russia  
Тел.: +7 (928) 196-21-67, e-mail: Visockaya\_anna@list.russi.ru



**Анастасия Александровна Белякова** – старший директор дирекции планирования и распределения продукции ОАО «Глория Джинс». Автор работ по проблеме оптимизации транспортных систем методами потокового программирования.

**Anastasia Aleksandrovna Belyakova** – senior director of «Gloria Jeans» stock company's directorate of planning and product distribution. Author's are dedicated to problem of transport systems' optimizing, using the methods of streaming programming.

346414, г. Ростов-на-Дону, ул. Портовая, д. 80/2, кв.42  
80/2 Portovaya st., app. 42, 346414, Rostov-na-Donu, Russia  
Тел.: +7 (928) 196-21-67, e-mail: Visockaya\_anna@list.ru

---

Федеральное агентство по образованию  
ГОУ ВПО «Сибирский Государственный

Технологический университет»  
Институт дополнительного профессионального образования, повышения квалификации  
и переподготовки специалистов

Центр тестирования профессионального образования

**III Всероссийская научно-практическая конференция  
с международным участием  
«Тестирование в сфере образования: проблемы и перспективы развития»**

**19–20 мая 2010 г.**

**г. Красноярск**

Направления работы конференции:

1. Информационные технологии в тестировании и обучении.
2. Инновационные технологии и педагогические измерения в образовании.
3. Управление качеством образования.
4. Психолого-педагогическое сопровождение личности в процессе тестирования.

**Контакты:**

660049, г. Красноярск, ул. Ленина, д. 71, ком. 205, ГОУ ВПО «Сибирский государственный технологический университет», оргкомитет конференции «Тестирование в сфере образования: проблемы и перспективы развития».

Дополнительную информацию можно получить по тел.: (391)265-29-26 (Данилова Ольга Сергеевна), и по электронной почте conf-test@mail.ru или fpkp@sibstu.kts.ru.