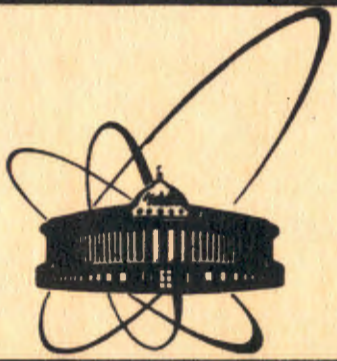

сообщения
объединенного
института
ядерных
исследований
дубна

Р11-85-349

Е.Ю.Мазепа, К.П.Муминов*, Е.Д.Федюнькин

АЛГОРИТМЫ ПЕРЕЧИСЛЕНИЯ СПЕЦИАЛЬНЫХ СУГРАФОВ И ИХ РЕАЛИЗАЦИЯ НА ЯЗЫКЕ "С"

* Научно-исследовательский институт прикладной физики ТашГУ

1985

1. СТЕКОВЫЕ АЛГОРИТМЫ ПЕРЕЧИСЛЕНИЯ ВСЕХ ν-ГРАФОВ ЗАДАННОГО СИММЕТРИЧЕСКОГО ГРАФА

Эти алгоритмы были предложены в работах [1,2] и используются, в основном, при решении задач дискретной оптимизации в тех случаях, когда требуется найти абсолютный оптимум. Сошлемся, например, на задачу оптимизации электрических сетей [3] /0 -графы/ и на задачу коммивояжера [4] /1-графы/.

Напомним, что ν-графом заданного симметрического графа $G=(X,U)$ /X - множество вершин, U - множество ребер/ называется суграф /по другой терминологии - частичный граф/ $G^{\nu}=(X,u)$, имеющий цикломатическое число ν и столько же компонент связности, сколько исходный граф $G=(X,U)$. Здесь $u \subset U$, т.е. $G^{\nu}=(X,u)$ получается исключением из графа $G=(X,U)$ некоторого числа ребер. Цикломатическим числом графа G с N вершинами, M ребрами, P компонентами связности называется число [5] $\nu(G)=M-N+P$; $\nu(G)$ совпадает с числом независимых циклов графа G. В частности, для дерева $\nu(G)=0$, а для графа, имеющего единственный цикл, $\nu(G)=1$.

Для того чтобы граф $G=(X,U)$ имел хотя бы один ν-граф $G^{\nu}=(X,u)$, нужно, чтобы цикломатическое число ν^* исходного графа $G=(X,U)$ удовлетворяло соотношению $\nu^* \geq \nu$. В частности, если $\nu^*=\nu$, то граф $G=(X,U)$ имеет единственный ν-граф $G^{\nu}=(X,u)$, и последний совпадает с исходным графом: $G^{\nu}(X,u)=G(X,U)$, $u=U$.

Стековые алгоритмы перечисления, по-видимому, наиболее эффективны, как в смысле быстродействия, так и в смысле необходимой памяти ЭВМ, поэтому именно они были выбраны для включения в состав инструментальной системы, предназначенной для работы с граф-информацией. Чтобы избежать усложнений по тривиальному поводу, мы будем рассматривать графы без петель.

Авторы пользуются случаем, чтобы исправить опечатку, которая имеется на рис.2 в работах [1,2]: блок $\boxed{F:=0}$, расположенный слева вверху в логической схеме функции $PT(G)$, должен быть заменен на блок $\boxed{F:=1}$. /В работе [1] этот блок имеет номер 8/.

2. МОДИФИЦИРОВАННАЯ ПРОЦЕДУРА КОНТРОЛЯ СВЯЗНОСТИ

Существенной частью стековых алгоритмов [1,2] является процедура контроля на связность $CT(G,W,m)$, которая проверяет, изменится ли число компонент связности после исключения из графа G ребра m /т.е. является ли ребро m перешейком/. Аргумент W - флаговый массив; если $W(i)=1$, то ребро i уже исключено из гра-

фа G. Процедура CT используется на каждом шаге стекового алгоритма и является его наиболее трудоемкой частью. При реализации алгоритма на ЭВМ она потребляет большую часть машинного времени. Чтобы было ясно дальнейшее, изложим основную идею процедуры CT.

Пусть G(1,m) и G(2,m) - граничные вершины ребра m. Если после исключения ребра m в графе существует цепь, соединяющая вершины G(1,m) и G(2,m) то исключение ребра m не изменит числа компонент связности. Такую цепь и ищет процедура CT.

Шаг 1. Пометим граничные вершины исключенного ребра двумя различными цветами. Пометим ребро m и исключенные ребра /для которых W(i) = 1/ как использованные.

Шаг 2. Обнулим флаговую переменную F. Сканируем список ребер, игнорируя помеченные. Если встретится ребро с окрашенной вершиной, переходим к шагу 3. Если список ребер исчерпан, переходим к шагу 5.

Шаг 3. Помечаем ребро, как использованное. Если вторая вершина окрашена, переходим к шагу 4. Окрашиваем вторую вершину в тот же цвет, в какой окрашена первая. Присваиваем F:=1 и возвращаемся к продолжению шага 2.

Шаг 4. Если граничные вершины одноцветные, возвращаемся к продолжению шага 2. Выход из процедуры с признаком "ребро m - не перешеек".

Шаг 5. Если F = 1, переходим к началу шага 2. Выход из процедуры с признаком "ребро m-перешеек".

Одинаковая окраска вершин означает, что их можно соединить цепью. Во время работы алгоритма окраска распространяется из граничных вершин ребра m, пока не появится пара смежных разноцветных вершин. Последнее означает, что существует еще одна цепь, соединяющая G(1,m) и G(2,m), т.е. ребро m не является перешейком.

Если за время очередного сканирования списка ребер флаг F остался равным 0 /т.е. распространение окраски не произошло/, то ребро m - перешеек/.

Модификация, которую мы предлагаем, сводится к следующему. Алгоритм CT вновь сканирует список ребер, если за время предыдущего сканирования произошло распространение хотя бы одного цвета. Между тем, легко показать, что, если за время предыдущего сканирования не произошло распространения обоих цветов, то ребро m - перешеек. Действительно, за время одного сканирования списка ребер гарантируется окраска всех вершин, смежных уже окрашенным. Если, например, распространение цвета G(1,m) не произошло, значит, исчерпаны все вершины, которые можно соединить цепью с G(1,m), и, следовательно, вершина G(2,m) недостижима.

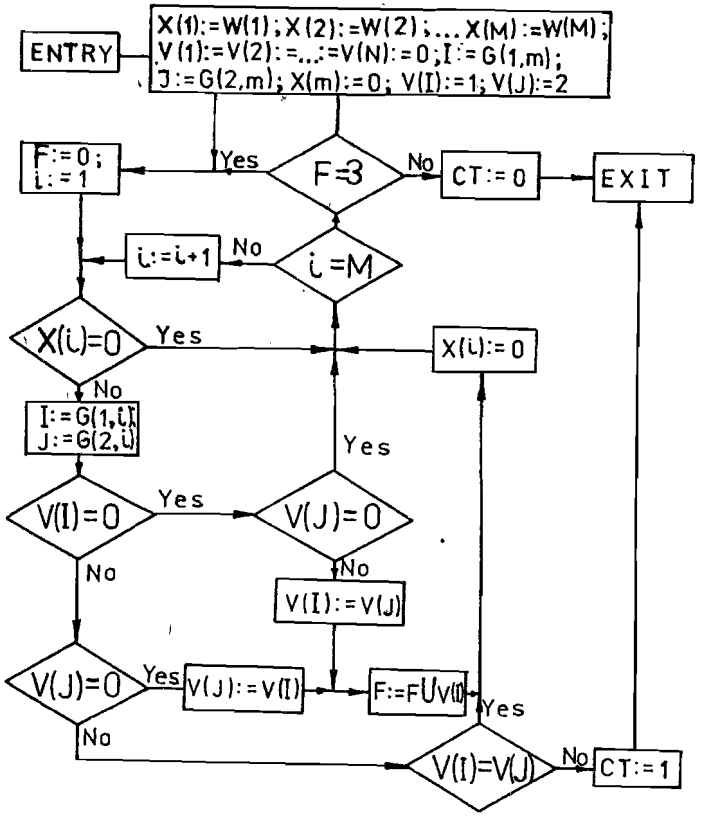

Рис.1. Модифицированная функция CT(G,W,m)- контроль связности.

Модифицированная функция CT(G,W,m) представлена на рис.1. В среднем она дает существенный выигрыш во времени. Здесь массив G(2,m) - список ребер исходного графа, представленных парами номеров своих граничных вершин. M - число ребер в графе. Массивы X(M) и V(N) - рабочие, N - число вершин в графе. I, J, i - рабочие переменные. F - рабочая флаговая переменная, F = 1 соответствует распространению цвета 1, F = 2 соответствует распространению цвета 2, F = 3 соответствует распространению обоих цветов. Через F:=FUV(I) обозначено логическое объединение F и V(I). Функция CT возвращает значение 0, если ребро m является перешейком и 1 - в противном случае.

3. АЛГОРИТМ ПЕРЕЧИСЛЕНИЯ ВСЕХ ГАМИЛЬТОНОВЫХ ЦИКЛОВ ГРАФА, ИСПОЛЬЗУЮЩИЙ ПРОЦЕДУРУ ПЕРЕЧИСЛЕНИЯ 1-ГРАФОВ

Напомним, что гамильтоновым циклом графа G называется цикл без самопересечений, проходящий через все вершины графа G.

Мы умеем получать все 1-графы графа G. Гамильтонов цикл тоже является 1-графом. Если дополнить алгоритм перечисления ν-графов процедурой проверки на "гамильтоновость", то получится алгоритм, перечисляющий все гамильтоновы циклы исходного графа G. Логическая схема такой процедуры изображена на рис.2. Обозначения те же, что и раньше. Ясно, что исходный граф должен быть связным. При конструировании процедуры HC мы воспользовались следующей простой леммой.

Лемма. Для того чтобы связный 1-граф был гамильтоновым циклом, необходимо и достаточно, чтобы степени его вершин не превышали 2.

Необходимость следует из того, что степени вершин любого цикла без самопересечения равны 2.

Достаточность будет доказана, если показать, что из отсутствия вершин степени больше 2 следует отсутствие вершин степени меньше 2. Поскольку граф односвязный, речь может идти только о вершинах степени единица. Пусть такая вершина существует. Поскольку 1-граф содержит цикл, должна существовать цепь, соединяющая эту вершину с циклом. В месте присоединения цепи к циклу должна существовать вершина степени не меньше трех, что противоречит условиям леммы.

Если исходный граф содержит перешейки, то он заведомо не содержит в себе гамильтоновых циклов, поэтому сначала полезно выяснить, существуют ли в исходном графе перешейки. Такое выяснение является попутным результатом деятельности процедуры NT в работах[1,2].

Отметим еще следующий неожиданный результат. Если перечислить ν-графы ν-связного исходного графа, проверяя каждый из них с помощью процедуры HC, то получим механизм для отыскания всех семейств гамильтоновых циклов исходного ν-связного графа /по одному гамильтонову циклу на компоненту связности/. Этот результат основывается на следующей теореме:

Теорема. Для того чтобы ν-связный ν-граф являлся семейством гамильтоновых циклов, необходимо и достаточно, чтобы степени его вершин не превышали 2.

Необходимость очевидна.

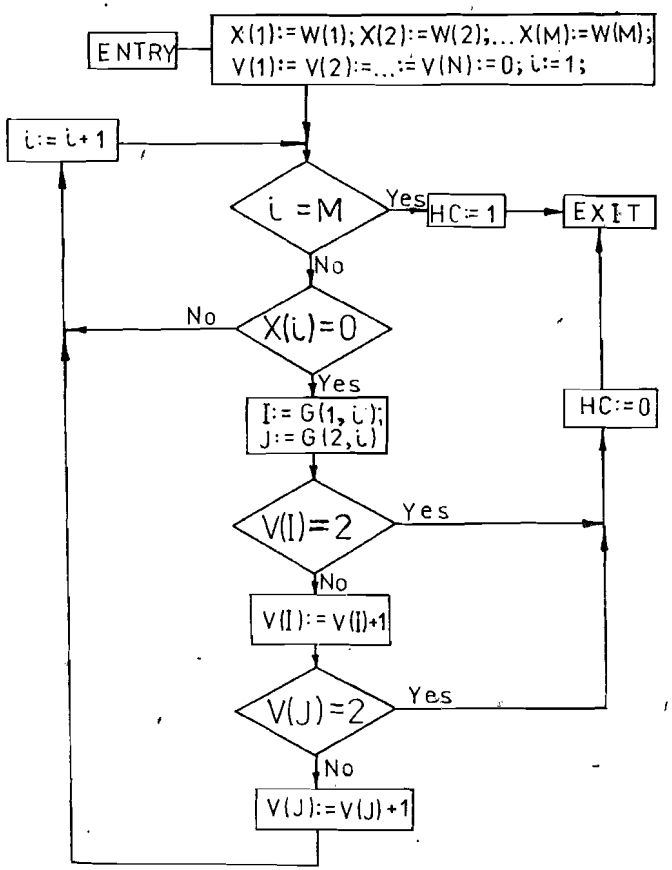

Рис.2. Функция HC(G,W) - проверка 1-графа на "гамильтоновость".

Достаточность. Если каждая из компонент связности содержит один цикл, то такая компонента является 1-графом. Справедливость теоремы следует из последовательного применения леммы к каждой из компонент. Если не каждая компонента связности содержит один цикл, то найдется, по крайней мере, одна компонента, содержащая больше одного независимого цикла. Такая компонента должна содержать вершину степени не меньше 3, что противоречит условиям теоремы.

4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ПЕРЕЧИСЛЕНИЯ НА ЯЗЫКЕ "С"

Язык "С" выбран авторами для создания инструментальной системы для работы с граф-информацией по двум причинам:

1. Существует широкий класс ЭВМ с операционными системами типа UNIX, которые поддерживают унифицированную операционную обстановку для программных продуктов, написанных на языке "С".

2. Язык "С" приспособлен для эффективной работы со сложно структурированной информацией.

В настоящее время реализовано два алгоритма перечисления ν-графов /основной и использующий понятие квазицикла/ и алгоритм перечисления гамильтоновых циклов /семейств гамильтоновых циклов/.

Важным вопросом при создании такого рода инструментальных систем является вопрос о способе представления информации. С одной стороны, должен существовать стандарт, обеспечивающий унифицированный входной язык для частных процедур. С другой стороны, должна быть обеспечена максимально возможная гибкость, обеспечивающая возможность настройки на разнотипную информацию. Укажем, например, что с ребрами и вершинами могут ассоциироваться имена, веса или даже массивы по-разному структурированной информации: все зависит от особенностей частной задачи. Должен быть обеспечен механизм для хранения граф-информации в базе данных и извлечения ее оттуда. Информация должна храниться в базе данных в инвариантном виде, а в памяти ЭВМ - в виде, наиболее удобном для использования. Авторы решили эту противоречивую задачу следующим образом.

Объект снабжается уникальным именем и уникальным идентификатором типа объекта.

Существуют три основных типа объекта:

1. HEADER, в котором содержится объединяющая информация о графе в целом.

2. Список вершин графа.

3. Список ребер графа.

Структура перечисленных выше объектов стандартизована. Структура других объектов зависит от особенностей частной задачи и не стандартизована. Все процедуры работы с графом имеют единственный параметр: адрес HEADER'а. В HEADER'е содержатся адреса массива вершин и массива ребер. В свою очередь, в массиве вершин и массиве ребер имеются ссылки на ассоциированные с ними объекты нестандартного типа.

5. НЕКОТОРЫЕ ЗАМЕЧАНИЯ О СТИЛЕ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ "С"

В отличие от таких языков, как FORTRAN и автокод, язык "С" мало приспособлен для прямого перевода на него с языка логических схем. По-видимому, мышление на языке логических схем и на языке "С" плохо совместимы. Часто оказывается, что проще непосредственно написать алгоритм на "С".

Приемы эффективного программирования на "С" не похожи на приемы эффективного программирования на других языках. Поскольку "С"-транслятор - оптимизирующий, следует писать программы так, чтобы не мешать процессу оптимизации. В частности, нужно стремиться к уменьшению числа помеченных операторов и к максимально свернутой записи. Эти особенности приводят к резкому увеличению чисто синтаксической деятельности в работе программиста.

Ретроспективный обзор существующих технологических систем приводит к выводу, что язык "С", вроде бы, нетехнологичен. На самом деле, технология содержится в нем самом, и нам кажется, что "С"-технология, несмотря на развитый поток управления, органически несовместима со структурным программированием, HIPO технологией и т.п. Заметим, кстати, что программирование "без GOTO" в "С" /6/ и в структурном подходе - понятия диспаратные.

Авторы признательны А.Бабаеву, М.Ю.Попову, В.Я.Фарисееву за полезные обсуждения; О.Н.Ломидзе - за содействие в работе и Ф.Ш.Хамраеву, который стимулировал интерес авторов к данной теме.

ЛИТЕРАТУРА

1. Мазепа Е.Ю., Силин И.Н., Федюнькин Е.Д. ОИЯИ, Р5-12874, Дубна, 1979.
2. Мазепа Е.Ю., Силин И.Н., Федюнькин Е.Д. Программирование, М., 1981, №4, с.78-86.
3. Сапожников А.П. ОИЯИ, 10-80-338, Дубна, 1980.
4. Н.Кристофидес. Теория графов. Алгоритмический подход. /Пер. с англ./. "Мир", М., 1978.
5. К.Берж. Теория графов и ее применения. /Пер. с англ./. ИЛ, М., 1962.
6. Kernighan W.Brain, Ritche M.Dennis. The "C" Programming Language. Englewood Liffs Prentice-Hall, 1978.

Рукопись поступила в издательский отдел
14 мая 1985 года.

Вниманию организаций и лиц, заинтересованных в получении публикаций Объединенного института ядерных исследований

Принимается подписка на препринты и сообщения Объединенного института ядерных исследований.

Установлена следующая стоимость подписки на 12 месяцев на издания ОИЯИ, включая пересылку, по отдельным тематическим категориям:

ИНДЕКС	ТЕМАТИКА	Цена подписки на год
1.	Экспериментальная физика высоких энергий	10 р. 80 коп.
2.	Теоретическая физика высоких энергий	17 р. 80 коп.
3.	Экспериментальная нейтронная физика	4 р. 80 коп.
4.	Теоретическая физика низких энергий	8 р. 80 коп.
5.	Математика	4 р. 80 коп.
6.	Ядерная спектроскопия и радиохимия	4 р. 80 коп.
7.	Физика тяжелых ионов	2 р. 85 коп.
8.	Криогеника	2 р. 85 коп.
9.	Ускорители	7 р. 80 коп.
10.	Автоматизация обработки экспериментальных данных	7 р. 80 коп.
11.	Вычислительная математика и техника	6 р. 80 коп.
12.	Химия	1 р. 70 коп.
13.	Техника физического эксперимента	8 р. 80 коп.
14.	Исследования твердых тел и жидкостей ядерными методами	1 р. 70 коп.
15.	Экспериментальная физика ядерных реакций при низких энергиях	1 р. 50 коп.
16.	Дозиметрия и физика защиты	1 р. 90 коп.
17.	Теория конденсированного состояния	6 р. 80 коп.
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники	2 р. 35 коп.
19.	Биофизика	1 р. 20 коп.

Подписка может быть оформлена с любого месяца текущего года.

По всем вопросам оформления подписки следует обращаться в издательский отдел ОИЯИ по адресу: 101000 Москва, Главпочтамт, п/я 79.

Мазепа Е.Ю., Муминов К.П., Федюнькин Е.Д. P11-85-349

Алгоритмы перечисления специальных суграфов и их реализация на языке "С"

Предложена модификация стекового алгоритма перечисления ν-графов, обладающая улучшенным быстродействием. Предложен алгоритм поиска гамильтоновых циклов, использующий стековый алгоритм перечисления ν-графов. Рассмотрена программная реализация указанных алгоритмов на языке "С". Обсуждаются особенности оптимального программирования на "С" для данного класса проблем.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1985

Перевод О.С.Виноградовой

Mazepa E.Yu., Muminov K.P., Fedyunkin E.D. P11-85-349

On the Generation of Special Spanning Subgraphs and Its "C" Language Realization

Fast modification of the ν-graphs generation algorithm is proposed. Hamilton cycles generation algorithm based on the previous algorithm is proposed. "C" language computer realization is described. "C" optimal programming features for such problems are discussed.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1985