

СПИСОК ЛИТЕРАТУРЫ

1. Черников С.Н. Линейные неравенства. — М.: Наука, 1968. — 488 с.
2. Вен В.Л., Литвинцев П.И., Фролов К.К. Решение системы линейных неравенств по методу наименьших квадратов // Журнал вычислительной математики и математической физики. — 1984. — Т. 24. — № 1. — С. 144–148.
3. Вылегжанин О.Н., Шкатова Г.И. Решение задачи линейного программирования с использованием оператора-проектора // Известия Томского политехнического университета. — 2009. — Т. 314. — № 5. — С. 37–40.
4. Вылегжанин О.Н., Шкатова Г.И. Учет ограничений равенств при решении оптимизационных задач с линейными ограничениями // Известия Томского политехнического университета. — 2008. — Т. 312. — № 5. — С. 76–78.

Поступила 13.03.2012 г.

УДК 004.94

МАТРИЧНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ РАЗРЕЗАНИЯ ДВУДОЛЬНЫХ ГРАФОВ

Ан.В. Погребной, Д.В. Погребной

Томский политехнический университет
E-mail: avpogrebnoy@gmail.com

Предложен алгоритм решения задачи разрезания двудольного графа на заданную совокупность минимально связанных подграфов. Алгоритм использует матричное представление двудольного графа и учитывает его специфику. Это позволило свести задачу разрезания к задаче линейного математического программирования. Алгоритм может работать как с произвольно сформированным, так и улучшенным исходным вариантом разрезания двудольного графа.

Ключевые слова:

Двудольный граф, исходное разрезание, матрица связности вершин графа, задача линейного математического программирования.

Key words:

Bipartite graph, the original cut, connectivity matrix of the vertices, problem of linear mathematical programming.

Введение

Двудольные графы [1] используют при построении графовых моделей, описывающих отношения между объектами разбитыми на два множества. В таких моделях отношения (рёбра) могут связывать объекты (вершины) только из разных множеств. Отношения между объектами внутри множеств отсутствуют или не учитываются. Примером графовых моделей, имеющих двудольную природу, являются сети Петри [2] с непересекающимися множествами позиций и переходов. Дуги в сети Петри могут связывать только переходы и позиции. Другим примером являются графы потока данных [3] с множествами модулей и данных, в которых дуги отражают потребление или формирование данных модулями.

Формально двудольный граф G описывается четвёркой (N, M, V, R) , где $N = \{e_i\}$, $i = 1, 2, \dots, n$; $M = \{e_j\}$, $j = 1, 2, \dots, m$ — множества вершин двудольного графа; $V = \{v_{ij}\}$ — множество рёбер v_{ij} , связывающих неупорядоченные пары вершин (e_i, e_j) , $e_i \in N$, $e_j \in M$; $R = \{r_{ij}\}$ — множество весов r_{ij} рёбер v_{ij} .

Если для всех $v_{ij} \in V$, $r_{ij} = 1$, то имеет место не взвешенный двудольный граф. Заметим также, что граф G рассматривается как неориентированный, т. к. пара вершин (e_i, e_j) , связанная ребром v_{ij} , принимается неупорядоченной. Ориентация дуг, например в графе потока данных, означает направление передачи данных. Вместе с тем, загрузка се-

ти при передаче данных по разрезанной дуге зависит от веса дуги и не зависит от её ориентации. Как в этом, так и во многих других примерах, ориентация дуг не учитывается и при решении задачи разрезания используется неориентированный взвешенный двудольный граф. Такой граф полностью описывается матрицей связности вершин $R = \|r_{ij}\|_{N \times M}$. Матрица R несимметрична и все её элементы (веса рёбер) $r_{ij} \geq 0$. Особенность матрицы R заключается в том, что она допускает перестановки строк R_i и столбцов R_j независимо друг от друга.

Задача разрезания возникает всякий раз, когда графовую модель, описывающую отношения между объектами распределённой системы, необходимо разбить на совокупность минимально связанных между собой подсистем. Если графовая модель представлена двудольным графом, то это, как правило, связано с наличием в системе объектов двух типов, которые нельзя смешивать в ходе решения задачи разрезания. В частности поэтому, для её решения нельзя использовать матричный алгоритм разрезания обыкновенного графа [4], в котором вершины не различаются по каким-либо признакам и образуют единое множество. Кроме того, двудольные графы в сравнении с обыкновенными обладают спецификой, которая даёт возможность разработать алгоритм разрезания, эффективно работающий для данного класса графов.

Предложенный в статье алгоритм можно отнести к алгоритмам последовательного типа. На первом этапе формируется исходный вариант разрезания графа G на подграфы G_q , $q=1,2,\dots,k$. Здесь фиксируется число k подграфов G_q , на которые разрезается граф G и для каждого q -го подграфа устанавливается число и состав вершин из множеств N и M . На последующих этапах выполняется улучшение разрезания за счёт изменения состава вершин в подграфах исходного разрезания $\{G_q\}$ так, чтобы сумма весов разрезанных рёбер была минимальной.

Алгоритм назван матричным, т. к. основан на представлении двудольного графа матрицей связности вершин R и при реализации поисковых процедур также использует матричное представление.

Формирование исходного варианта разрезания

Алгоритм разрезания будем рассматривать на примере взвешенного двудольного графа, представленного матрицей R на рис. 1, а, в котором множество N содержит 8, а M – 12 вершин. Граф G требуется разрезать на 3 подграфа G_q , $q=1, 2, 3$ с заданным числом вершин из множеств N и M .

На матрице R (рис. 1, а) вариант разрезания по аналогии с алгоритмом [4] можно представить совокупностью блоков, каждый из которых соответствует подграфу G_q с множеством вершин $N(G_q)$ и $M(G_q)$. В данном примере все блоки содержат по 4 вершины из множества M , а 8 вершин множества N распределены по блокам в количестве 3, 3, 2. Число блоков и их размерность, т. е. соотношение числа вершин из множеств N и M в соответствующих подграфах устанавливается пользователем. Последовательность расположения блоков разной размерности в матрице R формируется произвольно.

Элементы r_{ij} матрицы R , расположенные за пределами блоков, соответствуют межблочным связям, т. е. разрезанным рёбрам, связывающим разные подграфы. Эти элементы составляют часть матрицы, названную областью сечения. Таким образом, матрица R разбивается на область блоков и область сечения. Задача разрезания сводится к перестановкам строк R_i и столбцов R_j матрицы R

так, чтобы сумма элементов r_{ij} в области блоков была максимальной, а в области сечения – минимальной. Качество разрезания оценивается соотношением суммы элементов в блоках к сумме в области сечения. Чем больше величина этого соотношения, тем лучше вариант разрезания.

Приведённый на рис. 1, а, исходный вариант разрезания получен произвольным образом. Вершины $e_i \in N$ и $e_j \in M$ включены в очередной блок в порядке их нумерации. К тому же блок 3 в последовательности блоков мог быть поставлен первым или вторым. Соотношение сумм элементов в блоках и в области сечения для данного варианта разрезания равно 26/56. При визуальном анализе области блоков и области сечения легко обнаружить, что данный вариант разрезания можно улучшить. Так, например, если столбцы R_{i_1} и R_{i_2} поменять местами, то соотношение станет равным 35/47, т. е. улучшится на 9 единиц. Взаимная перестановка строк R_{i_1} и R_{i_2} также улучшает исходный вариант на 4 единицы.

Операции по поиску в множествах строк и столбцов перестановок, улучшающих качество разрезания, могут быть формализованы аналогично тому как это было сделано в [4]. Однако в данном случае более эффективным оказалось применение аппарата математического программирования. Такая возможность была найдена благодаря учёту отмеченной выше особенности матрицы R .

Метод улучшения качества разрезания

Рассмотрим подробнее операцию перестановки строк R_{i_1} и R_{i_2} . Каждая строка относительно блоков делится на подмножества элементов, принадлежащих столбцам соответствующих блоков $M(G_1)$, $M(G_2)$, $M(G_3)$. При перестановке строк элементы этих подмножеств всегда совместно попадают в некоторый блок или исключаются из него. В нашем случае при перемещении строки R_{i_1} на место R_{i_2} одно подмножество (0,1,2,0) исключается из 1-го блока, а другое (3,0,2,0) включается в 3-й блок. Аналогично для строки R_{i_2} при перемещении её на место R_{i_1} подмножество (0,0,3,1) включается в 1-й блок, а подмножество (0,1,0,1) исключается из 3-го блока.

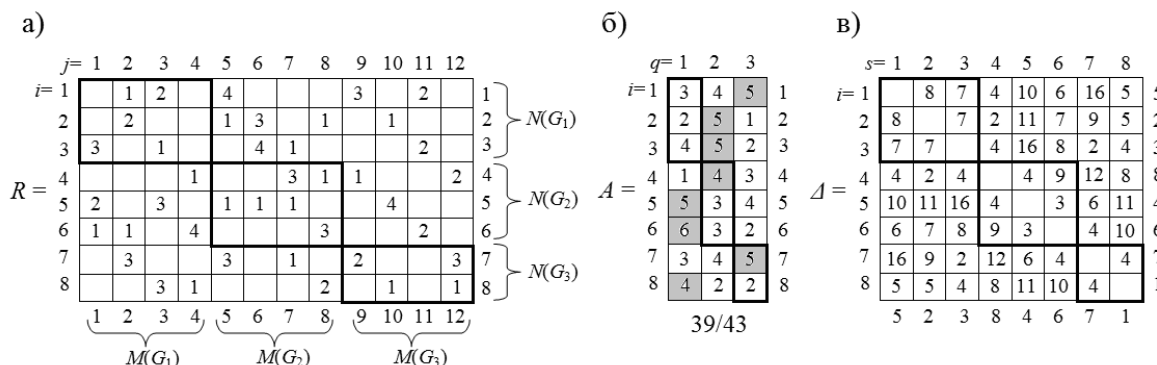


Рис. 1. Исходный вариант разрезания

Таким образом, исходя из цели решения задачи разрезания, при перестановке строк допустимо оперировать не подмножествами, а суммами элементов этих подмножеств. Сумму элементов r_{ij} строки R_i , относящихся к множеству $M(G_q)$, обозначим величиной α_{iq} ,

$$\alpha_{iq} = \sum_{r_{ij} \in M(G_q)} r_{ij}. \quad (1)$$

Величины α_{iq} вычисляются для всех строк и формируются в матрицу $A = \|\alpha_{iq}\|_{n \times k}$. Матрица A , полученная для исходного варианта разрезания (рис. 1, а), представлена на рис. 1, б. В итоге, поиск варианта распределения строк по блокам, улучшающего качество исходного разрезания, стало возможным вести на матрице A .

Матрица A устроена таким образом, что распределение i -й строки в q -й блок соответствует выбору в этой строке одного q -го элемента α_{iq} . При этом необходимо проследить, чтобы в каждом q -м столбце матрицы A было выбрано ровно n_q — число строк в множестве $N(G_q)$. Наибольшее улучшение исходного разрезания будет достигнуто при получении максимальной суммы выбранных элементов.

Для решения данной задачи введём переменную $x_{iq}=1$, если элемент α_{iq} выбран, и $x_{iq}=0$, в противном случае. Тогда задачу оптимального распределения строк по блокам можно записать в виде:

$$\sum_{i=1}^n \sum_{q=1}^k \alpha_{iq} x_{iq} \Rightarrow \max; \quad (2)$$

$$\sum_{i=1}^n x_{iq} = n_q, \quad q = 1, 2, \dots, k; \quad (3)$$

$$\sum_{q=1}^k x_{iq} = 1, \quad i = 1, 2, \dots, n. \quad (4)$$

Решив задачу (2)–(4), назовём её задачей TN , например венгерским алгоритмом [5], получим наилучшее распределение строк R_i по блокам при фиксированном положении столбцов. Результат решения задачи TN для матрицы A на рис. 1, б соот-

ветствует заштрихованным элементам α_{iq} . Из решения следует, что строки распределились по блокам следующим образом: $N(G_1)=(5,6,8)$, $N(G_2)=(2,3,4)$, $N(G_3)=(1,7)$. Качество данного разрезания составило 39/43.

Задача оптимального распределения столбцов по блокам решается для матрицы R , построенной с учётом перестановки строк согласно решению задачи TN . Полученная матрица R представлена на рис. 2, а. Справа от матрицы указана принадлежность строк к блокам. Последовательность расположения и размерность блоков в данной матрице сохраняется. Сформированное таким образом разрезание принимается в качестве исходного для решения задачи распределения столбцов.

Формализованная постановка данной задачи, названной задачей TM , осуществляется аналогично задаче TN . Вначале определяются элементы β_{qj} и формируются в матрицу $B = \|\beta_{qj}\|_{k \times m}$. Элементы β_{qj} вычисляются аналогично (1) по выражению:

$$\beta_{qj} = \sum_{r_{ij} \in N(G_q)} r_{ij}. \quad (5)$$

Матрица B , полученная для матрицы R (рис. 2, а), представлена на рис. 2, б.

Целевая функция и ограничения задачи TM записываются аналогично задаче TN :

$$\sum_{q=1}^k \sum_{j=1}^m \beta_{qj} x_{qj} \Rightarrow \max; \quad (6)$$

$$\sum_{j=1}^m x_{qj} = m_q, \quad q = 1, 2, \dots, k; \quad (7)$$

$$\sum_{q=1}^k x_{qj} = 1, \quad j = 1, 2, \dots, m. \quad (8)$$

Решение задачи (6)–(8) на матрице B отмечено штриховкой выбранных элементов β_{qj} . Качество данного варианта разрезания составило 56/26. На рис. 2, в, приведена матрица, полученная после перестановки столбцов согласно решению задачи TM . Принадлежность столбцов к блокам показана

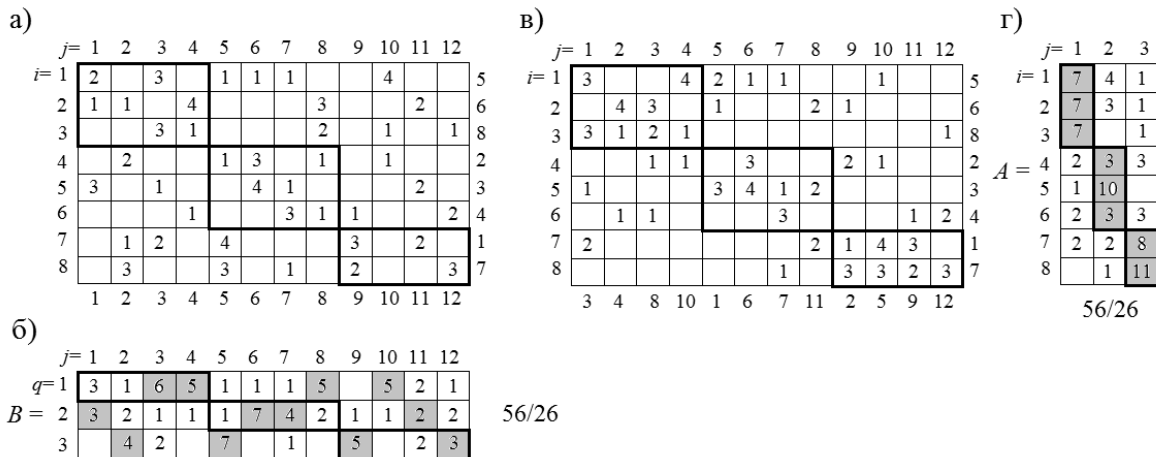


Рис. 2. Итоговый вариант разрезания

нумерацией снизу матрицы. Попытка улучшить для этой матрицы качество разрезания путём перестановки строк на основе решения задачи *TN* не привела к успеху. На рис. 2, *г*, видно, что в матрице *A* все строки сохранили свои места в блоках. Визуальный анализ области сечения матрицы на рис. 2, *в* также показывает, что путём перестановки какой-либо строки или столбца достигнутое качество разрезания нельзя улучшить. Ниже предлагается попытка достигнуть разрезание с более высоким качеством за счёт использования улучшенного исходного разрезания.

Методика поиска варианта исходного распределения строк по блокам.

Вариант разрезания с оценкой 56/26 получен на основе применения метода улучшения качества исходного разрезания, сформированного произвольным образом. В связи с этим возникает вопрос, имеется ли возможность улучшить качество разрезания, если исходное разрезание формировать не произвольным образом, а на основе некоторой процедуры рационального распределения строк или столбцов по блокам.

Рассматривая вопрос о предпочтениях в распределении строк по блокам, легко обнаружить, что в один блок должны попадать те строки, вершины которых соответствуют максимальным суммам весов связей с общими вершинами столбцов. В этом случае при попадании в общий блок данных столбцов и строк суммарный вес блока возрастает. Связность двух строк R_i и R_s , $i, s=1,2,\dots,n$, обозначим величиной δ_{is} , $i \neq s$. Значение величины δ_{is} определяется согласно выражениям:

$$\forall e_i, e_s \in N \ \& \ \forall e_j \in M[r_{ij} > 0 \ \& \ r_{sj} > 0] \Rightarrow (\delta_{isj} = r_{ij} + r_{sj}), \text{ иначе } (\delta_{isj} = 0); \quad (9)$$

$$\delta_{is} = \sum_{j=1}^m \delta_{isj}. \quad (10)$$

На рис. 3 приведены пояснения к вычислению δ_{is} на примере величины $\delta_{1,7}$. Вершины e_1 и e_7 , соответствующие строке R_{i_1} и R_{i_7} , связаны через общие вершины e_2, e_5, e_9 столбцов $R_{j_2}, R_{j_5}, R_{j_9}$, которые помечены штриховкой. Сумма весов r_{ij} и r_{sj} для общих вершин e_j , связывающих вершины e_1 и e_7 составила 16 единиц, т. е. $\delta_{1,7}=16$.

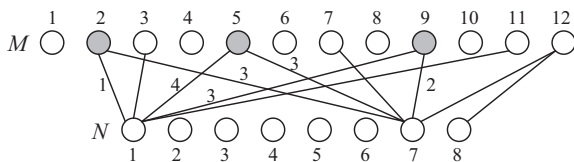


Рис. 3. Иллюстрация к определению $\delta_{1,7}$

Величины δ_{is} , вычисленные по выражениям (9), (10), формируются в матрицу связности строк Δ . Матрица Δ , полученная для исходной матрицы R (рис. 1, *а*), приведена на рис. 1, *в*. Для матрицы Δ с помощью матричного алгоритма (программное средство *CutGraf*) [6] решается задача разрезания, т. е. определяется распределение строк по блокам. При этом последовательность блоков и их размерность должна совпадать с этими параметрами в рассматриваемом примере, что и отражено на матрице Δ . Справа (и снизу) от матрицы указано полученное распределение строк по блокам.

Используя полученное распределение строк на матрице R формируется исходное разрезание и для него решается задача распределения столбцов, т. е. задача *TM*. Для нашего примера решение задачи *TM* привело к распределению столбцов $M(G_1)=(1,3,6,10)$, $M(G_2)=(4,7,8,11)$, $M(G_3)=(2,5,9,12)$. Оценка разрезания при данном распределении составила 58/24. Аналогичная методика формирования улучшенного исходного разрезания может быть применена и в отношении строк с последующим решением задачи *TN*. Достигнутое улучшение оценки на 2 единицы вовсе не означает, что с улучшенным исходным разрезанием всегда будет получено разрезание с более высокой оценкой.

Следует отметить, что после получения решения на основе произвольно сформированного исходного разрезания попытки поиска разрезания с более высокой оценкой являются вполне оправданными, т. к. не требуют больших вычислительных затрат. К таким попыткам, наряду с изложенной выше методикой улучшения исходного разрезания как относительно строк так и столбцов, относятся изменения последовательности решения задач *TN* и *TM*, а также изменения последовательности расположения блоков, имеющих разные размерности.

Заключение

1. Расширены возможности матричного алгоритма разрезания графов в направлении разработки более эффективного алгоритма решения этой задачи для двудольных графов. Учёт специфики двудольных графов сделал возможным применение в алгоритме улучшения разрезания графов этого типа методы линейного математического программирования.
2. Введение оценки связности вершин в одном из подмножеств двудольного графа через общие вершины другого подмножества позволило формировать исходное разрезание с более высокой оценкой. Для решения задачи поиска варианта распределения строк или столбцов по блокам исходного разрезания применен матричный алгоритм разрезания графов.

Работа выполнена в рамках государственного задания «Наука».

СПИСОК ЛИТЕРАТУРЫ

1. Зыков А.А. Основы теории графов. — М.: Вузовская книга, 2004. — 644 с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем. Пер. с англ. — М.: Мир, 1984. — 264 с.
3. Погребной В.К. Визуальный уровень представления алгоритмов функционирования распределённой системы реального времени на языке структурного моделирования // Известия Томского политехнического университета. — 2009. — Т. 314. — № 5. — С. 140–146.
4. Погребной В.К. Матричный алгоритм решения задачи разрезания графов // Известия Томского политехнического университета. — 2007. — Т. 310. — № 5. — С. 91–96.
5. Галкина В.Н. Дискретная математика: комбинаторная оптимизация на графах. — М.: Гелиос АРВ, 2003. — 232 с.
6. Погребной А.В., Погребной Д.В. Исследование матричного алгоритма решения задачи разрезания графов // Молодёжь и современные информационные технологии: Труды VIII Всерос. научно-практ. конф. молодых ученых. — Томск, 2010. — С. 140–141.

Поступила 27.02.2012 г.

УДК 519.673

АДРЕСУЕМАЯ ЯЧЕЙКА ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СТРУКТУРЫ ДЛЯ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

А.А. Хамухин

Томский политехнический университет
E-mail: aaxtru@tpu.ru

Предложена функциональная схема адресуемой ячейки однородной вычислительной структуры для решения дифференциальных уравнений в частных производных. Показана возможность ускорения решения за счет снижения длительности цикла загрузки данных в регистры ячеек. Представлены сравнительные результаты моделирования линейки адресуемых и не адресуемых ячеек на примере решения тестовой краевой задачи с постоянными и переменными коэффициентами.

Ключевые слова:

Адресуемая ячейка, однородная вычислительная структура, параллельные вычисления, дифференциальные уравнения в частных производных, постоянные и переменные коэффициенты.

Key words:

Addressable cell, homogeneous computing structure, parallel computation, partial differential equations, independent and dependent coefficients.

Введение

В статьях [1–3] описаны примеры реализации вычислительных устройств для решения дифференциальных уравнений в частных производных на основе одинаковых ячеек однородной вычислительной структуры (ОВС), показаны возможности реконfigurирования ОВС в соответствии с заданными граничными условиями без аппаратных добавлений в схему ячейки, приведены дополнения, необходимые для решения задач с переменными коэффициентами. Основу такой ячейки составляют сумматоры и масштабные умножители, а информация о коэффициентах дифференциальных уравнений хранится в нескольких регистрах каждой ячейки. Разработан ряд схем реализации таких ячеек, на которые получены патенты [4–6].

Главным преимуществом однородной вычислительной структуры является синхронное аппаратное распараллеливание решения дифференциального уравнения, которое позволяет добиться наибольшей производительности на данном классе задач при прочих равных условиях по сравнению с другими типами вычислительных устройств. Вторым немаловажным преимуществом ОВС являются относительно низкие аппаратные затраты в пе-

ресчете на одну операцию по сравнению с универсальным многоядерным микропроцессором, в котором большая часть транзисторов просто простаивает при решении данного класса задач.

В этих статьях также сделано предположение, что реализация ОВС возможна на современных микросхемах типа БМК (известных за рубежом под аббревиатурой *ASIC*) или ПЛИС (известных за рубежом под аббревиатурой *FPGA*).

Наиболее известные реализованные проекты в этой области — это платформа *RC100* фирмы *SGI* и суперкомпьютер фирмы *Cray*. Платформа *RC100* содержит 35 модулей, каждый из которых имеет по две ПЛИС *Altera Stratix III* и 10 блоков локальной памяти. По данным фирмы *SGI* платформа решает задачи биоинформатики в 900 раз быстрее, чем кластер из 68 узлов на базе универсальных микропроцессоров *AMD Opteron* [7].

Суперкомпьютер *Cray XT5h* объединяет в единую систему скалярные процессоры (на основе *AMD Opteron*) и векторные процессоры на основе ПЛИС. Он комплектуется из серверов двух типов: *Cray X2* и *Cray XR1*. Последний состоит из двух узлов, в каждый из которых входит микропроцессор *AMD Opteron*, связанный с помощью высокоско-