УДК 621.396

МЕТОД РЕШЕНИЯ ЗАДАЧИ О МИНИМАЛЬНОМ ПОКРЫТИИ КАК СРЕДСТВО ПЛАНИРОВАНИЯ В GRID

В.С. Пономаренко[(1)], С.В. Листровой[(2)]

[(1)] *Харьковский национальный экономический университет;*
[(2)] *Украинская академия железнодорожного транспорта, г. Харьков*

Предложен метод решения задач о наименьшем покрытии и наименьшем вершинном покрытии в произвольных графах, позволяющий строить эффективные алгоритмы решения задачи определения минимального числа кластеров, позволяющего решить в GRID заданное подмножество задач с требуемой эффективностью.

ВВЕДЕНИЕ

Глобальные вычислительные сети GRID [1] были предложены в качестве новой парадигмы для решения крупномасштабных вычислительных задач в науке, технике и бизнесе [2]. Они дают возможность одновременного использования многочисленных вычислительных ресурсов [3], принадлежащих различным организациям и расположенных в различных административных регионах. Системы GRID объединяют разнородные вычислительные ресурсы (персональные компьютеры, рабочие станции, кластеры, суперкомпьютеры), используя разные стратегии доступа, выполняя различные приложения (научные, инженерные и коммерческие), предъявляющие к системе различные требования. Ресурсы принадлежат различным организациям, имеющим свои правила управления, использования и определения их стоимости для различных пользователей в различное время. Доступность и загруженность ресурсов также может динамически изменяться во времени. Последние работы в области GRID позволяют приложениям использовать вычислительные ресурсы, принадлежащие различным организациям, распределенным по различным странам и континентам. Один из видов ресурсов GRID — однородные многопроцессорные системы (кластеры), которые могут состоять из сотен или даже тысяч процессоров. В процессе распределения ресурсов в GRID возникает необходимость в определении минимального числа кластеров, на которых можно выполнить заданное подмножество задач.

Рассмотрим двухуровневую систему, в которой на первом уровне несколько независимых брокеров распределяют вычислительные задачи на кластеры, а на втором уровне каждый кластер распределяет задачи, присвоенные ему локальным планировщиком. Пусть имеется n кластеров и каждый i-й процессор в кластере в состоянии решить некоторое подмножество L_i задач с требуемой эффективностью. Положим, что на решение поступило m задач, которые нужно решить. Требуется определить минимальное число кластеров, обеспечивающее решение всех m задач, как правило, $m > n$. Пусть возможности по решению задач определены графом, приведенным на рис. 1, кластеры

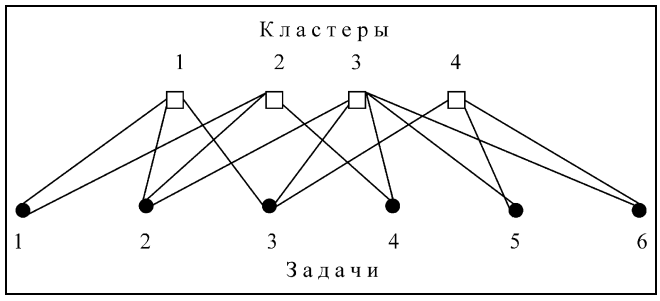

Рис. 1. Граф, отображающий возможности решения задач

соединены ребрами с теми задачами, которые они могут выполнить с требуемой эффективностью.

Требуется найти минимальное число кластеров, обеспечивающих выполнение всех задач с требуемой эффективностью. Представим граф (см. рис. 1) булевой матрицей B, в которой строкам соответствуют задачи, а столбцам — кластеры. Элемент (i, j) матрицы будем считать равным 1, если j-й кластер способен с требуемой эффективностью обеспечить решение i-й задачи, и равным нулю — в противном случае:

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}.$$

В данном случае задача сводится к определению минимального числа столбцов в матрице B, покрывающих единицами все строки в ней. Эта же задача может быть сформулирована как задача линейного булевого программирования, постановка которой в общем виде имеет вид:

$$L = \sum_{j=1}^{n} c_j x_j \to \min \qquad (1)$$

при ограничениях

$$\sum_{j=1}^{n} \beta_{ij} x_j \geqslant 1, \quad i = \overline{1, m}, \quad x_j \in \{0, 1\}; \quad c_j \geqslant 0, \qquad (2)$$

где

$$\beta_{ij} = \begin{cases} 1, \text{ если } i\text{-я переменная может быть} \\ \qquad \text{покрыта переменной } x_j, \\ 0 \text{ — в противном случае.} \end{cases} \qquad (3)$$

Задачи о наименьшем покрытии (ЗНП) и о наименьшем вершинном покрытии (ЗНВП) имеют широкое прикладное значение в теории построения сложных систем, в системах диагностики вычислительных систем и сетей [4], при разработке их программного и математического обеспечения, а также для планирования распределения ресурсов в GRID. Основное требование к алгоритмам решения данных задач состоит в высокой оперативности решения и обеспечении минимально возможной погрешности этих решений.

Частным случаем ЗНП является задача о наименьшем разбиении (ЗНР), которая получается из задачи (1)—(3) путем замены неравенства (2) на равенство. Вследствие особой природы задачи часто при ее исследовании удается сделать хорошо известные заранее выводы и упрощения [4, 5]: например, если в строке матрицы B присутствует одна единица, то столбец, в котором находится эта единица, обязательно принадлежит минимальному покрытию; если в матрице B есть столбцы, частично совпадающие с другими, но покрывающие меньшее число строк, то они могут быть исключены из анализа, так как любое множество, которое покрывает столбец с большим числом единиц, покрывает и все множества, которые покрывает столбец с меньшим числом единиц. Таким образом, в этом случае столбец с большим числом единиц доминирует над столбцом с меньшим числом единиц.

Ряд методов сначала разрабатывался для решения ЗНР, а потом адаптировался к решению в работах [6, 7]. В работе [1] предложены методы решения ЗНП, в которых используется дерево поиска и линейное программирование. Подходы, базирующиеся на рассмотрении отсекающих плоскостей и подобные в принципе тем, которые применяются в общем 0-1-программировании [8], представлены в работах [4, 9]. Сравнение этих методов и исследование их вычислительных характеристик приведено в работе [4]. Наиболее полный обзор методов решения ЗНП как задачи линейного булевого программирования, дан в статье [10], где показано, что алгоритмы на основе идей рангового подхода [15, 16] отличаются от методов, основанных на идеях метода ветвей и границ, меньшими временной сложностью и погрешностью.

Задачу нахождения независимых максимальных множеств или вершинных покрытий можно, например, решить последовательным перебором независимых множеств с одновременной проверкой каждого множества на максимальность (последнее осуществляется добавлением к исследуемому множеству дополнительной вершины, не принадлежащей ему, и выяснением, сохраняется ли независимость) и запоминанием максимальных множеств.

Однако с увеличением числа вершин этот способ становится весьма громоздким. На основе усовершенствования этой процедуры построены алгоритмы Брона и Кэрбоша [4]. Как показано в работах [13, 14], задача вершинного покрытия является NP-полной, и эффективные алгоритмы ее решения для произвольных графов неизвестны. Для двудольных графов на основе алгоритмов Хопкрофта и Карпа (с поиском в глубину) разработаны методы [11], позволяющие находить минимальное вершинное покрытие и максимальное независимое множество вершин в произвольном двудольном графе $H = \langle X, Y, E \rangle$ за время $O((m + n)\sqrt{n})$, где $n = |X \cup Y|$ и $m = |E|$. Полиномиальные алгоритмы

вычисления числа устойчивости были получены для совершенных графов — графов, у которых для любого его порожденного подграфа хроматическое число равно кликовому числу. Алгоритм вычисления числа устойчивости графа [16] основан на методе эллипсоидов и использует процедуру отделения матриц графа. В вычислительном плане этот алгоритм обладает рядом существенных недостатков, не позволяющих использовать его на практике. Как показано в работе [16], получить правильное решение при числе вершин в графе более 10-ти практически невозможно. Применение *r*-алгоритмов позволило увеличить размерность решаемых задач до 50, а при использовании двойственных оценок в схеме ветвей и границ — до 100, с погрешностью, не превышающей 5 % [16].

Таким образом, поскольку размерности задач, решаемых в GRID-системах, достаточно велики, представляется актуальной разработка эффективного приближенного алгоритма, решающего с единых позиций ЗНВП и ЗНП и обладающего малой временной сложностью и погрешностью.

1. ПОСТАНОВКА И МЕТОД РЕШЕНИЯ ЗАДАЧИ

Для решения с единых позиций ЗНВП и ЗНП будем представлять как граф $G(X, E)$, так и произвольную матрицу B в виде булевой функции. Рассмотрим конъюнктивное представление булевой матрицы B. Пусть задана булева матрица B с n столбцами и m строками. Столбцы будем задавать вектором $X = \{x_1, x_2, ..., x_n\}$, а строки — вектором $M = \{\mu_1, \mu_2, ..., \mu_m\}$. Покрытием Q строк M матрицы B назовем такое множество столбцов B, которое покрывает единицами все строки M. Для определения всех покрытий матрицы будем применять алгебраический метод получения по импликантной таблице приведенных систем простых импликант булевых функций. Если каждый столбец из совокупности $X = \{x_1, x_2, ..., x_n\}$ рассматривать как «простую импликанту», покрывающую совокупность строк $M = \{\mu_1, \mu_2, ..., \mu_m\}$, каждую строку μ_i как набор переменных, покрываемых простыми импликантами, то матрицу B можно представить как импликантную таблицу булевой функции. При такой интерпретации матрицы B для каждой строки μ_i можно записать дизъюнкцию столбцов b_i, покрывающих рассматриваемую строку, в следующем виде:

$$\begin{aligned} d\mu_I &= (X_1 \vee X_k \vee ...) \\ &... \\ d\mu_I &= (X_p \vee X_t \vee ...). \end{aligned} \qquad (4)$$

Конъюнкция дизъюнкций (4) по всем строкам $\mu_1, \mu_2, ..., \mu_m$ матрицы B образует конъюнктивное представление матрицы B, содержащее в себе все покрытия совокупности строк M:

$$\begin{aligned} k(M) &= d\mu_1 \cdot d\mu_2 \cdot ... \cdot d\mu_r = \\ &= (X_1 \vee X_k \vee ...) \cdot ... \cdot (X_p \vee X_t \vee ...). \end{aligned} \qquad (5)$$

Раскрывая скобки в соответствии с законами дистрибутивности, получаем дизъюнктивное представление матрицы B, образующее перечень всех возможных покрытий совокупности строк $M = \{\mu_1, \mu_2, ..., \mu_m\}$. Так, в соответствии с выражением (5) конъюнктивное представление матрицы

$$B = \begin{array}{c} \\ 1\text{–}2 \\ 1\text{–}3 \\ 1\text{–}4 \\ 2\text{–}7 \\ 3\text{–}6 \\ 4\text{–}5 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \left[\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}\right] \end{array},$$

задающей граф G (рис. 2), имеет вид:

$$\begin{aligned} F = (X_1 \vee X_2)(X_1 \vee X_3)(X_1 \vee X_4)(X_2 \vee X_7) \times \\ \times (X_3 \vee X_6)(X_4 \vee X_5). \end{aligned} \qquad (6)$$

Как показано в работе [17], если f — булева функция, построенная по графу $G = (V, E)$ в виде произведения дизъюнктов $(v_i \vee v_j)$, где $\{v_i\} \in \{0, 1\}$, $i = \overline{(1, n)}$, $j = \overline{(1, n)}$, $i \neq j$, и при этом каждый дизъюнкт $(v_i \vee v_j)$ соответствует ребру (v_i, v_j), то все наборы переменных $\{v_i, v_j\}$, на которых она принимает значение «истинно», соответствуют вершинным покрытиям в графе $G = (V, E)$. И для перечисления всех вершинных покрытий графа $G = (V, E)$ необходимо определить те системы зна-

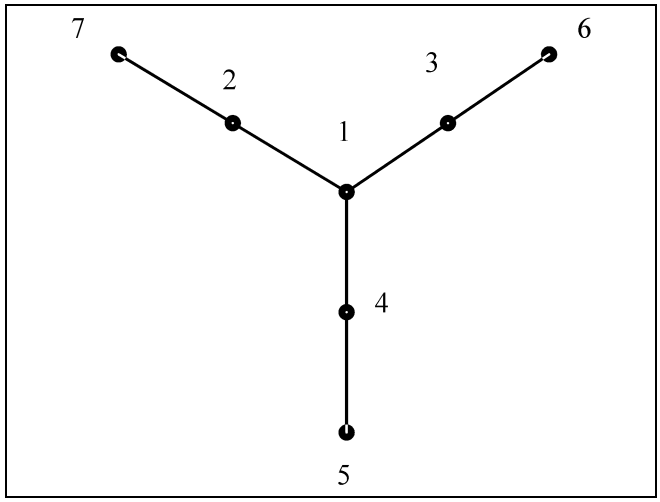

Рис. 2. Граф *G*

чений переменных $\{v_i, v_j\}$, при которых высказывание

$$F(V_1, V_2, ..., V_n) = 1 \qquad (7)$$

«истинно». Чтобы найти эти системы значений переменных $\{v_i, v_j\}$, необходимо привести левую часть выражения (7) к минимальной дизъюнктивной нормальной форме, раскрывая скобки и пользуясь законом поглощения. Такая форма единственная, ввиду отсутствия в выражении (7) логических отрицаний.

Введем следующие понятия и определения для элементов булевой матрицы B, задающей некоторый граф $G(X, E)$. Если в матрице B выделить произвольный столбец j, то с ним можно связать некоторое подмножество столбцов $\{q\}$, с которыми данный столбец пересекается, поскольку столбец j соответствует вершине графа, а число единиц в столбце определяется степенью вершины d_j, и подмножество $\{q\}$ будет всегда содержать d_j таких столбцов. Будем говорить, что подмножество столбцов $\{q\}_j$ образует связку столбцов относительно столбца j. Число столбцов в матрице B равно числу вершин n в графе $G(X, E)$, а число возможных связок столбцов для матрицы B равно n. Отметим, что все столбцы в матрице B пересекаются друг с другом только в одной строке, поскольку в каждой строке матрицы B находится по две единицы. Поэтому, с точки зрения их пересекаемости, возможность их принадлежности минимальному покрытию равновелика. Каждая связка столбцов $\{q\}_j$ покрывает определенное число строк l_j. Связку $\{q\}_j$, покрывающую максимальное число строк в матрице B, назовем максимальной. Если в графе $G(X, E)$ есть висячие вершины, то это означает, что в матрице B есть столбцы P, содержащие только одну единицу, и, следовательно, связки этих столбцов содержат только по одному столбцу в каждой. Эти столбцы обладают интересным свойством, которое для произвольных графов $G(X, E)$ определяет следующее

Утверждение. *Если граф $G(X, E)$ содержит некоторое подмножество висячих вершин $Q \in X$, то подмножество вершин $P \in X$ смежных с Q может быть дополнено до одного из минимальных вершинных покрытий графа $G(X, E)$.*♦

Доказательство. Пусть граф $G(X, E)$ содержит висячую вершину j, которая соединена с ним ребром (i, j). Тогда граф $G(X, E)$ можно представить в виде объединения подграфов G' и G'', при этом вершина i является точкой сочленения, т. е. принадлежит подграфам G' и G''. Пусть подграф G' содержит k вершин, тогда степени вершин в подграфе G' могут изменяться от 1 до $k - 1$. Предположим, что в графе $G(X, E)$ покрытие $\{X_i\}_{\min}$ не включает в себя вершину i, но тогда оно должно включать в себя вершину j (т. е. $j \in \{X_i\}_{\min}$) для того, чтобы ребро (i, j) оказалось покрытым. Пусть степень вершины i в подграфе G' равна 1 и вершина i в подграфе G' соединена с вершиной X_1. Поскольку ребро (i, X_1) должно быть покрыто, а вершина i в покрытие не входит, то $X_1 \in \{X_i\}_{\min}$. Следовательно, обе вершины $(j, X_1) \in \{X_i\}_{\min}$. Если вместо вершин (j, X_1) в покрытие ввести вершину i, то мы получаем новое покрытие $\{X_i\}'$, которое содержит на одну вершину меньше, чем покрытие $\{X_i\}_{\min}$. Это противоречит первоначальному предположению о том, что покрытие $\{X_i\}_{\min}$, содержащее вершину j, является минимальным вершинным покрытием в графе $G(X, E)$. Аналогичное рассуждение можно провести, полагая степени вершины i в подграфе G' равными 2, 3, ..., $k - 1$; следовательно, по индукции следует, что во всех возможных случаях возникает противоречие с первоначальным предположением, и вершина i принадлежит минимальному вершинному покрытию. Поскольку аналогичное рассуждение можно провести для любой висячей вершины графа $G(X, E)$, то по индукции утверждение можно считать доказанным.♦

В общем случае, если в графе существует несколько минимальных вершинных покрытий, то могут существовать и минимальные вершинные покрытия без вершины, смежной с висячей вершиной графа $G(X, E)$, но если оно одно, то эта вершина обязательно в нем присутствует. Таким образом, если в матрице B есть столбец j, содержащий одну единицу, то связка столбцов относительно столбца j (в данном случае она состоит из одного столбца) принадлежит минимальному покрытию $X_{\min}$. В случае представления графа в виде булевой функции $F(X_1, X_2, ..., X_n)$ данное утверждение эквивалентно тому, что если в дизъюнктах есть переменная X_p, которая в них встречается один раз, то переменная X_n, составляющая ей пару, в дизъюнкции входит в покрытие. И, следовательно, данную дизъюнкцию можно заменить переменной X_n, при этом исключить из анализа все дизъюнкции, содержащие переменную X_n. В работе [8] при анализе графов $G(X, E)$, не имеющих висячих вершин, представленных в конъюнктивной нормальной форме в виде некоторой булевой функции F, для определения наименьшего множества переменных $\{X_i\}$, которые покрывают все дизъюнкты в конъюнктивном представлении графа $G(X, E)$, предложено применять принцип су-

перпозиции в булевой алгебре, основанный на следующем равенстве:

$$f(X_1, X_2,..., X_n) = f(X_1 = 0, X_2, ..., X_n) \vee \vee f(X_1, X_2 = 0, ..., X_n) \vee ... \vee f(X_1, X_2, ..., X_n = 0). \quad (8)$$

Не нарушая принципа суперпозиции, соотношение (8) можно представить в виде

$$f(X_1, X_2, ..., X_n) = X_1 \cdot f(X_1, X_2, ..., X_n) \vee \vee X_2 \cdot f(X_1, X_2, ..., X_n) \vee ... \vee X_n \cdot f(X_1, X_2, ..., X_n). \quad (9)$$

Особенность конъюнктивного представления графа $G(X, E)$ в виде булевой функции состоит в том, что она содержит число дизъюнктов, равное числу ребер в графе, а число переменных в каждом дизъюнкте равно 2, и каждая переменная соответствует некоторой вершине графа $G(X, E)$. Введем понятие характеристического вектора $h_q = (h^{j_1}_{i=1}, h^{j_2}_{i=2}, ..., h^{j_n}_{i=n})_q$ некоторой булевой функции

$$f_q = X_p \cdot X_h \cdot ... \cdot X_q f(X_1, X_2, ..., X_n), \quad (10)$$

в которой переменные X_p, X_h, ..., X_q не встречаются в функции $f(X_1, X_2, ..., X_n)$.

Вес j_i в векторе h_q указывает, как часто переменная X_i встречается в дизъюнктах функции $f(X_1, X_2, ..., X_n)$, а сам вектор будем описывать суммарной весовой характеристикой

$$V_q = \sum_{i=1}^{n} j_i. \quad (11)$$

Рассмотрим алгоритм решения данной задачи в виде следующей *процедуры A* преобразования булевой функции $f(X_1, X_2, ..., X_n)$, задающей некоторый граф $G(X, E)$.

Шаг 1. Проверяем, есть ли в дизъюнктах $f(X_1, X_2, ..., X_n)$ переменные $\{X_k\}$, встречающиеся по одному разу. Если да, то умножаем $f(X_1, X_2, ..., X_n)$ на переменные $\{X_l\}$, стоящие совместно в дизъюнктах с переменными $\{X_k\}$ и образующие сомножитель $X_p \cdot X_h \cdot ... \cdot X_q$, состоящий из r переменных, находившихся в дизъюнктах совместно с X_k, при этом все дизъюнкты в $f(X_1, X_2, ..., X_n)$, содержащие переменные X_p, X_h, ..., X_q, исключаем из дальнейшего анализа.

Шаг 2. Проверяем, встречаются ли в полученной функции $f = X_p \cdot X_h \cdot ... \cdot X_q \cdot f(X_1, X_2, ..., X_n)$ оставшиеся переменные по одному разу или нет; если да, то мы получили минимальное вершинное покрытие, в которое входят вершины, соответствующие переменным $X_p \cdot X_h \cdot ... \cdot X_q$ и переменным, выбираемым по одной из каждого оставшегося дизъюнкта, и процедура заканчивается; если нет, то выполняем следующий шаг.

Шаг 3. Полученные функции $f = X_p \cdot X_h \cdot ... \cdot X_q \cdot f(X_1, X_2, ..., X_n)$ умножаем поочередно на переменные X_i, оставшиеся в $f(X_1, X_2,..., X_n)$, и исключаем при этом дизъюнкты, содержащие X_j, если их q, то мы получим q таких функций $\{f_1, f_2, ..., f_q\}$, и для каждой функции вычисляем характеристический вектор $h_q = (h^{j_1}_{i=1}, h^{j_2}_{i=2}, ..., h^{j_n}_{i=n})_q$ и его весовую характеристику $V_q = \sum_{i=1}^{n} j_i$. Среди них выбираем функцию f_q с минимальным значением весовой характеристики и переходим к шагу 1. Если же все весовые характеристики функций окажутся одинаковыми, то выбираем любую из них. ♦

Данная процедура представляет собой «жадный» алгоритм, который строит минимальное вершинное покрытие в графе $G(X, E)$ заданным в виде конъюнктивного представления матрицы B, в которой столбцам i соответствуют вершины графа $\{X_i\}$, а строкам s — ребра $(X_q, X_j) \in E$ графа.

В процессе выполнения алгоритма, когда мы полагаем $X_i = 1$, появляется произведение X_pX_k, ..., X_c, которое поглощает все дизъюнкты, содержащие эти переменные, последнее эквивалентно удалению из графа вершин (i, p, k, ..., c) и ребер, им инцидентных. В результате такого преобразования исходного графа возможны два варианта: вновь полученный подграф может содержать или не содержать висячие вершины.

Если подграф содержит висячую вершину, то, включая ее в минимальное покрытие и удаляя из графа вместе с инцидентными ей ребрами, получим новый подграф. Если в процессе преобразования мы каждый раз получаем подграф с висячими вершинами, то в соответствии с утверждением мы получим в худшем случае за n шагов минимальное вершинное покрытие исходного графа. Так, если исходный граф является деревом, содержащим n ярусов и имеющим ширину, равную h (под шириной дерева подразумевается максимально возможное число вершин на ярусе дерева), то данная стратегия последовательного включения в покрытие вершин, смежных с висячими, позволит построить минимальное вершинное покрытие дерева за $O(hn)$ шагов.

В случае, когда после удаления вершин (i, p, k, ..., c) и ребер, им инцидентных, получается подграф, не содержащий висячих вершин, можно выделить три варианта: два предельных случая, когда получаемый подграф является полно связанным компонентом или образует простой цикл на остав-

шемся множестве вершин, и оптимальность работы алгоритма в этих крайних случаях очевидна; промежуточный вариант — когда в подграфе нет висячих вершин и степени его вершин произвольны. Обоснование оптимальности работы алгоритма в этом случае дает следующая

Теорема. *Если при $X_j = 1$ получается связка столбцов $\{q\}_j^*$, покрывающая максимальное число строк в матрице B, то применение к ней процедуры A приводит к построению минимального покрытия матрицы B.*♦

Доказательство. Предположим, что множество столбцов, принадлежащее связке $\{q\}_j^*$, покрывает максимально возможное из всех связок столбцов число строк l, а оставшиеся $(m - l)$ строк непокрытые. Процедура A, добавляющая число столбцов до полного покрытия строк матрицы B, на каждом шаге добавляет в покрытие каждый раз столбец, покрывающий максимальное число строк из непокрытых. Следовательно, к столбцам $\{q\}_j^*$ добавится минимальное число столбцов $\{p\}$. Итак, мы получили покрытие, состоящее из множества столбцов $\{q\}_j^* \cup \{p\}$. Предположим, что оно не минимально, но поскольку число добавленных до покрытия столбцов $\{p\}$ процедурой A минимально, последнее возможно, если существует связка столбцов, а связку столбцов $\{q\}_j$ покрывает число строк $k > l$. Это противоречит первоначальному предположению о том, что $\{q\}_j^*$ является максимальной связкой столбцов, следовательно наше предположение неверно и множество столбцов образует минимальное покрытие матрицы B графа $G(X, E)$.♦

Потеря оптимального решения при работе процедуры A происходит тогда, когда возникает ситуация, при которой весовые характеристики всех булевых функций f_q в процедуре A оказываются одинаковыми, и тогда мы выбираем любую из них. Однако возникновение такой ситуации возможно в основном на однородных графах, что в задачах планирования ресурсов в GRID встречается довольно редко. Ясно, что процедура A может быть применена не только для булевых матриц, которые задают некоторый граф $G(X, E)$, но и для произвольных булевых матриц.

Пример. Определим минимальное вершинное покрытие в графе, приведенном на рис. 3.

Строим булеву функцию графа:

$$f = (x_1 \vee x_2)(x_1 \vee x_3)(x_1 \vee x_5)(x_2 \vee x_3) \times \\ \times (x_2 \vee x_6)(x_3 \vee x_4)(x_4 \vee x_5)(x_5 \vee x_6).$$

Определяем, есть ли переменные, которые встречаются один раз. Если да, то умножаем функцию на переменную, соседнюю с ней, иначе выписываем все функции, умноженные поочередно на все переменные, присутствующие в ней, и вычисляем векторы, характеризующие данные функции и их весовые характеристики:

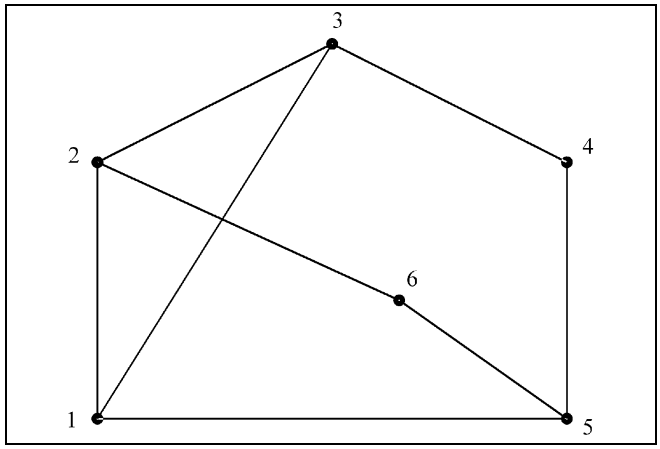

Рис. 3. Граф *G*

$$f_1 = x_1(x_2 \vee x_3)(x_2 \vee x_6)(x_3 \vee x_4)(x_4 \vee x_5)(x_5 \vee x_6);$$
$$h_1 = (2_2, 1_3, 2_4, 2_5, 2_6); \quad V_1 = 9;$$

$$f_2 = x_2(x_1 \vee x_3)(x_1 \vee x_5)(x_3 \vee x_4)(x_4 \vee x_5)(x_5 \vee x_6);$$
$$h_2 = (2_1, 2_3, 2_4, 3_5, 1_6); \quad V_2 = 10;$$

$$f_3 = x_3(x_1 \vee x_5)(x_2 \vee x_6)(x_4 \vee x_5)(x_5 \vee x_6);$$
$$h_3 = (2_1, 2_2, 1_4, 2_5, 2_6); \quad V_3 = 9;$$

$$f_4 = x_4(x_1 \vee x_2)(x_1 \vee x_3)(x_1 \vee x_5)(x_2 \vee x_3)(x_2 \vee x_3) \times \\ \times (x_5 \vee x_6); \quad h_4 = (3_1, 3_2, 2_3, 2_5, 2_6); \quad V_2 = 12;$$

$$f_5 = x_5(x_1 \vee x_2)(x_1 \vee x_3)(x_2 \vee x_3)(x_2 \vee x_6)(x_3 \vee x_4);$$
$$h_5 = (3_1, 2_2, 2_3, 1_4, 1_6); \quad V_5 = 9^*;$$

$$f_6 = x_6(x_1 \vee x_2)(x_1 \vee x_3)(x_1 \vee x_5)(x_2 \vee x_3)(x_3 \vee x_4) \times \\ \times (x_4 \vee x_5); \quad h_6 = (3_1, 2_2, 3_3, 2_4, 2_5); \quad V_5 = 12.$$

Из полученных таким образом функций выбираем ту, в которой суммарная весовая характеристика вектора минимальна: функция f_5. Поскольку в выражении для f_5 переменная x_4 встречается только один раз, умножаем f_5 на переменную соседнюю с ней — это x_3 — и получаем:

$$f_{53} = x_5x_3(x_1 \vee x_2)(x_2 \vee x_6).$$

Поскольку в выражении для f_{53} переменные x_6 и x_1 встречается только один раз, то умножаем f_5 на переменную соседнюю с ней — это x_2 — и получаем:

$$f_{532} = x_5x_3x_2.$$

Таким образом, минимальное вершинное покрытие образуют вершины {2, 3, 5}.

2. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

Исследование временной сложности разработанного алгоритма проводилось для произвольных графов с различными плотностями ребер в графе. Плотность A изменялась от 0,1 до 0,5, а число N вершин — от 4 до 100. Графики зависимости числа Q элементарных операций от числа вершин в графе приведены на рис. 4. Как видно, временная сложность алгоритма определения минимальных вершинных покрытий в графе в среднем не превышает величины $O(0{,}9n^3)$.

Все результаты получены с доверительной вероятностью 0,95, погрешность решений не превышала 2—6 %, а процент неточных решений не превышал 20 %.

ЗАКЛЮЧЕНИЕ

Предложенный метод позволяет с достаточно высокой оперативностью и точностью решать как задачи о наименьшем покрытии, так и задачи о наименьшем вершинном покрытии. Временную сложность соответствующего алгоритма можно легко понизить до $O(n^2)$, если его реализовывать на n-процессорной вычислительной системе, поскольку функции f_q при реализации процедуры A можно вычислять одновременно независимо друг от друга, что очень важно для планирования распределением ресурсов в системах, работающих в реальном времени, к которым и относятся системы распределения ресурсов в GRID. Данный метод даст возможность эффективно решать динамически изменяющуюся задачу определения минимального числа кластеров, позволяющих решить заданное подмножество задач с требуемой эффективностью.

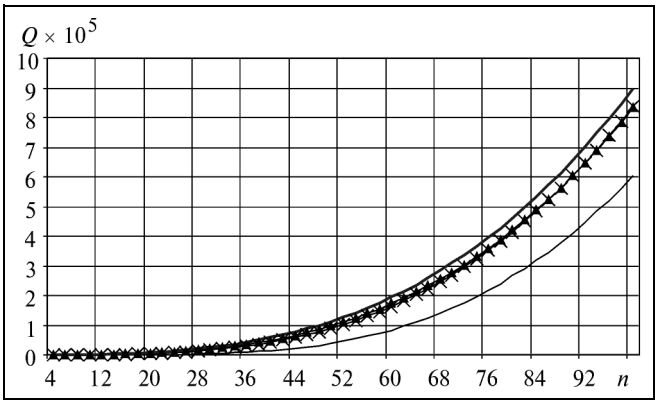

Рис. 4. Зависимость числа элементарных операций от размерности графа при различных значениях плотностей ребер в графе: ——— — $A = 0{,}1$; −×− — $A = 0{,}3$; −▲− — $A = 0{,}5$; ——— — $A = 0{,}9n^3$

ЛИТЕРАТУРА

1. *Brucker P.* Scheduling Algorithms. — Springer Ferlag, 1998. — P. 217—218.
2. *Methods* and Experiences of Parallelizing Flood Models / L. Hluchy, et al. // The 10th EuroPVM/MPI Conference. LNCS 2840. Sept. 2003, Venice. — P. 677—681.
3. *Baker B.S., Brown D.J. and Katseff H.P.* A 5/4 algorithm for two-dimensional packing // Jornal of Algorithms. — 1981. — Vol. 2, — P. 348—368.
4. *Кристофидес Н.* Теория графов. Алгоритмический подход. — М.: Мир, 1978. — 309 с.
5. *Balinski M.* Integer programming: methods, uses, computation // Man. Sci. — 1965. — N 12. — P. 253.
6. *Garfinkel R.S., Nemhauser G.L.* The set partitioning problem: set covering with equality constraints // Ops. Res. — 1969. — N 17. — P. 848.
7. *Pierce J.F.* Application of combinatorial programming to a class of all-zero-one integer programming problems // Man. Sci. — 1968. — N 15. — P. 191.
8. *Gomory R.* An algorithm for integer solutions to linear programs, Recent Advances in mathematical Programming. — N-Y.: McGraw-Hill, 1963.
9. *Bellmore M., Ratliff H.D.* Set covering and involuntary bases // Man. Sci. — 1971. — N 18. — P. 427.
10. *Листровой С.В., Симашкевич О.Н.* Об использовании гарантированных прогнозов в методах решения задач булевого программирования на основе рангового подхода // Электронное моделирование. — 2003. — Т. 25. — № 4. — С. 89—103.
11. *Липский В.* Комбинаторика для программистов. — М.: Мир, 1988. — 203 с.
12. *Шор Н.З., Стеценко С.И.* Квадратичные экстремальные задачи и недифференцируемая оптимизация. — Киев: Наукова думка, 1989. — 196 с.
13. *Пападимитриу К., Стайглиц М.* Комбинаторная оптимизация. Алгоритмы и сложность. — М.: Мир, 1985. — 512 с.
14. *Гэри М., Джонсон Д.* Вычислительные машины и трудноразрешаемые задачи. — М.: Мир, 1982. — 416 с.
15. *Листровой С.В., Гуль А.Ю.* Метод решения задачи о минимальном покрытии на основе рангового подхода // Электронное моделирование. — 1999. — № 1. — С. 58—70.
16. *Listrovoy S.V. and Gul A.Yu.* Method of Minimum Covering Problem Solution on the Basis of Rank Approach // Engineering Simulation. — 1999. — Vol. 17. — P. 73—89.
17. *Листровой С.В., Яблочков С.В.* Метод решения задачи определения минимальных вершинных покрытий и независимых максимальных множеств // Электронное моделирование. — 2003. — Т. 25. — № 2. — С. 31—43.

☎ *+38057-730-10-62, e-mail: om1@yandex.ru*

Статья представлена к публикации членом редколлегии В.Д. Малюгиным. ❑