Поиск минимальных покрытий булевой матрицы с использованием параллельных вычислений

Дюкова Е. В., Инякин А. С., Нефёдов В. Ю.
djukova@ccas.ru
Москва, Вычислительный центр РАН

В ряде случаев при построении процедур распознавания и кластеризации используется аппарат дискретной математики, в частности методы поиска покрытий булевой или целочисленной матрицы (например, при конструировании алгоритмов вычисления оценок, при синтезе элементарных классификаторов в логических процедурах распознавания, при построении логического корректора на базе элементарных классификаторов). Среди разного вида задач поиска покрытий одной из центральных является задача поиска минимальных покрытий булевой матрицы.

Пусть L — булева матрица. Набор столбцов H матрицы L называется *покрытием*, если каждая строка матрицы L в пересечении хотя бы с одним из столбцов, входящих в H, даёт единицу. Покрытие называется *неприводимым*, если никакое его собственное подмножество покрытием не является. Покрытие минимальной длины называется *минимальным* покрытием. Требуется найти все минимальные покрытия матрицы L.

Данная задача является трудной в вычислительном плане [1], поэтому важна разработка эффективных (в практическом плане) алгоритмов её решения.

Поскольку все минимальные покрытия являются неприводимыми, то для их поиска можно адаптировать алгоритмы нахождения неприводимых покрытий. В настоящее время разработан целый ряд таких алгоритмов [2, 3, 4, 5], среди которых следует выделить асимптотически оптимальные алгоритмы. Асимптотически оптимальный алгоритм работает с полиномиальной задержкой на каждом шаге, и число его шагов асимптотически равно числу всех неприводимых покрытий для почти всех булевых матриц размера $m \times n$ при условии, что $n \to \infty$ и $m^{\alpha} \leqslant n \leqslant 2^{m^{\beta}}$, $\alpha > 1$, $\beta > 1$.

В работе рассмотрен асимптотически оптимальный алгоритм поиска неприводимых покрытий из [4] (алгоритм A). Данный алгоритм основан на нахождении на каждом шаге набора столбцов H, содержащего максимальную единичную (перестановочную) подматрицу. Если при этом набор столбцов H не содержит нулевой строки, то он является неприводимым покрытием. Алгоритм работает с полиномиальной задержкой $O\left(q^2mn\right)$, где $q = \min(m, n)$.

Реализованы и экспериментально исследованы три модификации алгоритма A для поиска минимальных покрытий булевой матрицы. Первые

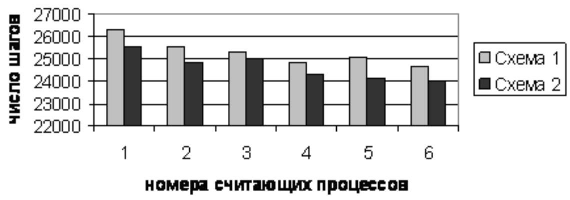

Рис. 1. Загруженность процессов.

два алгоритма, а именно, алгоритм A_1 и алгоритм A_2, осуществляют односторонний обход дерева решений. Третий алгоритм (алгоритм A_3) осуществляет обход дерева решений широким фронтом. Алгоритм A_1 использует в качестве первоначальной верхней оценки длины минимального покрытия оценку, выдаваемую жадным алгоритмом (алгоритмом наискорейшего спуска), а алгоритм A_2 — оценку, выдаваемую «улучшенным» жадным алгоритмом.

В отличие от классического жадного алгоритма, «улучшенный» жадный алгоритм взвешивает единичные элементы.

Для оценки эффективности алгоритмов A_1, A_2 и A_3 проведён ряд экспериментов на однопроцессорной ЭВМ, которые показали, что «улучшенный» жадный алгоритм позволяет ускорить решение задачи в среднем на 5–10% и сократить число шагов алгоритма. Под шагом алгоритма понимается построение очередной висячей вершины в дереве решений. Эксперименты также показали, что при больших размерах матриц обход широким фронтом невыгоден.

Алгоритм A_2, показавший наилучшие результаты, адаптирован для использования на многопроцессорных комплексах. При этом реализованы две схемы распараллеливания. В обеих схемах присутствует выделенный управляющий процесс, который раздаёт задания считающим процессам. В первой схеме в случае нахождения покрытия меньшей длины, чем текущая верхняя оценка длины минимального покрытия, считающий процесс сообщает об этом управляющему, который рассылает информацию всем остальным. Во второй схеме считающий процесс сам рассылает новую оценку всем остальным считающим процессам.

Важными показателями качества распараллеливания являются равномерность загрузки считающих процессов (Рис. 1) и масштабируемоть, т. е. зависимость времени счёта от числа считающих процессов. Счёт на случайных матрицах показал, что при использовании как первой,

Рис. 2. Масштабируемость.

так и второй схемы все считающие процессы загружены достаточно равномерно. Обе реализованные схемы показали типичную для задач дискретной оптимизации масштабируемость (Рис. 2). Вторая схема имеет небольшое преимущество по времени счёта и числу шагов.

Работа выполнена при поддержке РФФИ, проекты № 07-01-00516 и № 05-01-00495.

Литература

[1] *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.

[2] *Дюкова Е. В.* О сложности реализации дискретных (логических) процедур распознавания // ЖВМиМФ. — 2004. — Т. 44, № 3. — С. 550–572.

[3] *Дюкова Е. В.* Об асимптотически оптимальном алгоритме построения тупиковых тестов // ДАН СССР. — 1977. — Т. 233, № 4. — С. 527–530.

[4] *Дюкова Е. В., Инякин А. С.* О сложности решения задачи построения поиска неприводимых покрытий булевой матрицы. — М.: ВЦ РАН, 2006. — 24 с.

[5] *Инякин А. С.* Алгоритмы поиска неприводимых покрытий булевой матрицы. — М.: ВЦ РАН, 2004. — 25 с.