

М.И. Абрамский, В.В. Кирюхин

Севастопольский национальный технический университет

Студгородок, г. Севастополь, Украина, 99053

E-mail: abramsky@inbox.ru

ОЦЕНКА БЫСТРОДЕЙСТВИЯ ЭКСПРЕСС-АЛГОРИТМОВ МАРШРУТИЗАЦИИ НА ГРАФОВЫХ МОДЕЛЯХ

Предлагаются параллельные версии алгоритмов отыскания экстремальных подграфов (кратчайшие пути, минимальный и максимальные остовы) на графе, отображающем структуру геоинформационной системы. Оценена их эффективность по критерию выигрыша во времени реализации по сравнению с их последовательными аналогами.

Одной из основных задач на графе, отображающем структуру некоторой сложной системы, например, геоинформационной (ГИС), является поиск оптимального маршрута, т.е. пути между двумя вершинами, характеризующимся экстремальным значением некоторого показателя (например, пути минимальной длины).

Системы можно классифицировать на статические и динамические.

Статические ГИС, например, отличаются неизменностью ландшафта и связанной с ним информации. Для статических систем не очень важна скорость обработки графа, так как рассчитывается он только один раз, а затем информация повторно используется. Примером статической системы может быть ГИС, которая представляет упрощенную модель движения поездов, в которой все маршруты заранее известны, равно как и график движения поездов. Упрощенная в том смысле, что не принимаются во внимание различные случайные факторы, такие как: форс-мажорные обстоятельства, выход из строя поездов и самого железнодорожного полотна и т.п.

Для динамических систем очень важна скорость решения задач на графе. Особенно она важна для геоинформационных систем, в которые включены подсистемы управления транспортными потоками на автомобильных магистралях и улицах крупных городов, управления воздушным движением и т.п. Помимо скорости начального (базового) расчета, важным фактором в системах реального времени является скорость последующих пересчетов графа в связи с изменением окружающей обстановки.

В связи с тем, что динамические ГИС обычно используются крупными предприятиями или организациями, в дальнейшем будем исходить из предположения, что финансовые затраты на аппаратное обеспечение достаточны для решения поставленной задачи. Поэтому традиционные последо-

вательные алгоритмы оптимизации здесь могут быть заменены их параллельными аналогами и версиями, рассчитанными на использование параллельных мультипроцессорных систем. Подчеркнем, что здесь речь идет именно о параллельных МПС, а не о специализированных процессорах типа ассоциативных, рассматриваемых, например, в работе [1].

Как и во многих других задачах, использование параллельных алгоритмов для работы с ГИС дает значительный выигрыш в скорости при условии достаточных аппаратных мощностей с сохранением того же уровня качества решения. В дальнейшем будем называть подобные алгоритмы *экспресс-алгоритмами*, подчеркивая тем самым основную их особенность.

Структура ГИС обычно отображается некоторым графом (I, W) , где $I = \dots \{i\}$ – множество вершин графа, соответствующих характерным точкам (объектам) ГИС, и $W = \{(i, j)\}$ – множество дуг; дуга (i, j) соединяет вершины i и j графа и имеет некоторый вес $R(i, j)$, чаще всего интерпретируемый как расстояние между объектами i и j .

Одной из основных задач на графе ГИС является задача отыскания кратчайшего пути между двумя заданными вершинами, скажем, u_1 и u_n . Существует множество алгоритмов решения этой задачи. Из них наиболее приемлемым для параллельного выполнения нам представляется алгоритм Дейкстры [2]. Будем называть экспресс-алгоритмом алгоритм, реализующий некоторый (возможно, параллельный) метод вычислений без ограничения на количество вычислителей (процессоров).

Ниже предлагается экспресс-алгоритм (ЭАД) отыскания кратчайшего пути на базе метода Дейкстры.

Начало

1. Вводим матрицу $R = \{R(i, j)\}$, $i, j \in I$.
2. Полагаем $d(i) = GM$, $i \neq u_1$; $d(u_1) = 0$; здесь GM – максимальное число, представляемое в процессоре.
3. Всем вершинам $i \in I$ присваиваем метки $m(i)$, полагая сначала $m(i) = 0$, $i \in I$.
4. Вершину u_1 объявляем текущей: $t = u_1$;
5. Для всех вершин, у которых $m(i) = 0$, пересчитываем вес по формуле: $d(i) := \min \{d(i), d(t) + R(t, i)\}$;
6. Среди вершин, для которых $m(i) = 0$, ищем ту, для которой вес $d(i)$ минимален; если минимум не найден, т.е. если вес всех не "помеченных" вершин равен GM , то путь не существует; переход на Конец.
7. Найденную вершину с минимальным весом полагаем текущей и помечаем ее ($m(t) = 1$).
8. Повторяем п. 6 для всех остальных аналогичных вершин, распределяя каждое очередное задание на новый процессор.

9. Если $t = u_2$ на каком-либо из процессоров, то найден путь длины $d(t)$; выводим информацию о кратчайшем пути; переходим на Конец.

10. Освобождаем процессоры, переходим к п. 5.

Конец.

Эффективность экспресс-алгоритма удобно оценивать его быстродействием. Здесь под быстродействием будем понимать коэффициент ускорения, определяемый как отношение K времени реализации алгоритма на однопроцессорной ВС ко времени его реализации на мультипроцессорной ВС без ограничения числа процессоров. При этом полагаем, что сам экспресс-алгоритм представляется ярусно-параллельной формой (ЯПФ) [3] максимальной ширины, что как раз соответствует ситуации "полного распараллеливания", т.е. отсутствию ограничений на число процессоров. Если эти процессоры характеризуются одинаковым быстродействием, то K можно определить как отношение суммы всех блоков алгоритма к длине ЯПФ; в свою очередь под длиной здесь понимается сложность вычислений, или просто сложность.

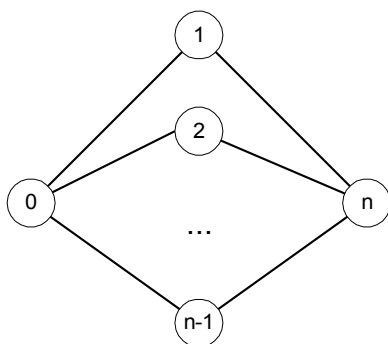


Рисунок 1 – Пример графа, для которого сложность ЭАД равна 2

Заметим, что последний термин сейчас широко используется именно в указанном смысле, хотя в классическом понимании "сложность алгоритма" есть количество информации, необходимое для описания алгоритма. Однако, как мы надеемся, контекст однозначно позволит избежать терминологической путаницы.

Сложность оригинального алгоритма Дейкстры для графа с n вершинами составляет n^2 [3]. Сложность ЯПФ экспресс-алгоритма ЭАД может составлять 2 – например, для графа на рисунке 1.

Максимальная сложность ЯПФ равна n , соответствующий граф показан на рисунке 2.

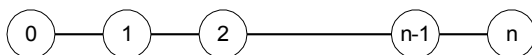


Рисунок 2 – Пример графа, для которого сложность ЯПФ ЭАД равна n

В остальных случаях сложность ЯПФ алгоритма ЭАД варьируется от 2 до n . Следовательно, эффективность K экспресс-алгоритма ЭАД лежит в пределах $n \leq K \leq n^2/2$.

Рассмотрим, какое влияние может оказать применение алгоритма ЭАД и других экспресс-алгоритмов на время T_p реакции ГИС [4]. В общем случае это время можно оценить по формуле

$$T_p = T_{cd} + T_{nz} + T_{alg} + T_{вр}, \quad (1)$$

где T_{cd} – время на считывание (получение) данных; T_{nz} – время, требуемое на проведение всех необходимых расчетов по подготовке графа (расчет весов дуг); T_{alg} – время работы алгоритма поиска кратчайшего пути на графе; $T_{вр}$ – время на выдачу результата.

Пренебрежем временем на считывание данных и выдачу результата (в хорошо отлаженных системах эти времена весьма малы). Оценим сложность подготовительных расчетов, т.е. время T_{nz} . Для графа размером n , при наличии k информационных массивов [5], это время будет порядка:

$$T_{nz} = (k^2 + 1) \cdot n^2. \quad (2)$$

Так как сложность оригинального алгоритма Дейкстры оценивается как n^2 , то $T_{alg} = n^2$.

Тогда

$$T_p \approx T_{alg} + T_{nz} \approx (k^2 + 2) \cdot n^2. \quad (3)$$

Примем, что средняя сложность ЯПФ алгоритма ЭАД равна $n/2$, тогда время T_{alg}^* на его реализацию составит $n/2$.

Применение экспресс-алгоритмов к подготовительному этапу и методик распараллеливания позволяют получить:

- обобщенную информационную матрицу [4] (суперпозицию информационных массивов) за 1 квант времени при наличии достаточного числа $(k^2 \cdot n^2)$ элементарных вычислительных единиц;
- таблицу весов путем функционального соединения информации о рельефе и обобщенной информационной матрицы [5] за 1 квант.

Тогда $T_{nz}^* = 2$ и

$$T_p^* \approx T_{nz}^* + T_{alg}^* \approx 2 + \frac{n}{2}. \quad (4)$$

С учетом (3) и (4) эффективность K системы вышеуказанных экспресс-алгоритмов составит

$$K = \frac{T_p}{T_p^*} = \frac{2 \cdot (k^2 + 2) \cdot n^2}{2 + n}.$$

Если $k \gg 1$, $n \gg 1$, что имеет место в реальных ГИС, то

$$K \approx 2 \cdot k^2 \cdot n. \quad (5)$$

Как видно из (5), применение экспресс-алгоритмов дает значительный выигрыш в быстродействии по сравнению с последовательными алгоритмами.

В качестве одного из самых простых, но эффективных алгоритмов построения минимального остова будем использовать методику Прима [2]. Ниже приведено описание параллельного алгоритма Прима (ЭАП).

Начало

1. Вводим матрицу $R = \{R(i, j)\}$, $i, j \in I$. Квадратная матрица R представляет собой первоначальный граф. Порядок матрицы R равен n .
2. Всем процессорам, число которых равно числу вершин, ставится в соответствие вершина (или «полюс» в нотации Прима). Каждому процессору дается задание выбрать кратчайшую связь (v, n) между ассоциированной вершиной и другой вершиной, причем $(v, u) \in R$. В результате мы получим подграфы, представленные матрицами $V_s = \{V_s(k, m)\}$, где $s = 1..n/2$.
3. На все процессоры подается задание попарного соединения подграфов V_s кратчайшими связями. $V_1 \cup V_2, V_3 \cup V_4, \dots, V_{s-1} \cup V_s$ — для случая, когда s — четное. В противном случае один граф ни с чем не соединяется. В результате s уменьшится в два раза для четного s или до $(s-1)/2 + 1$ в случае нечетного значения s .
4. Выполняем п.3, пока s не равно 1. Если s равно 1, значит минимальный остов построен и его отображение содержится в матрице V_1 .

Конец.

Сложность алгоритма Прима составляет n^2 [2]. Проанализируем сложность алгоритма ЭАП. В общем случае она равна $2n-1$, однако для частного случая сложность может составлять и 1. Следовательно, эффективность K алгоритма ЭАП лежит в пределах $n^2/(2n-1) \leq K \leq (n^2/2)$. Оценим время реакции системы при использовании непараллельного алгоритма. Для графа размером n , при наличии k информационных массивов с учетом (1) и (2) оценка будет иметь вид:

$$T_p \sim T_{n2} + T_{алг} \sim (k^2 + 2) \cdot n^2. \quad (6)$$

Для параллельной реализации, с учетом ее сложности, при тех же условиях, что и в (6), однако с учетом предыдущих замечаний по распараллеливанию подготовительного этапа T_{n2} , получим следующую оценку времени реакции системы:

$$T_p^* \sim T_{n2}^* + T_{алг}^* \sim 2 + (2 \cdot n - 1) = 2 \cdot n + 1. \quad (7)$$

Эффективность K системы определяется из (6) и (7) и равна

$$K = \frac{T_p}{T_p^*} = \frac{(k^2 + 2) \cdot n^2}{2 \cdot n + 1}. \quad (8)$$

При выборе приближения авторы руководствовались теми же предположениями о значениях n и k , что и для параллельного алгоритма Дейкстры.

Из анализа формулы (8) очевидно, что использование распараллеливания значительно уменьшает время работы системы.

В статье рассматриваются методики и алгоритмы, а также оценка их эффективности, в задачах поиска оптимального маршрута и других задачах на графах динамических геоинформационных систем, в которых параметры дуг графа, а также сама его структура могут меняться с быстротой, сравнимой со скоростью пересчета.

Предложены параллельные версии известных алгоритмов отыскания экстремальных подграфов на некотором графе, который является моделью геоинформационной системы.

Эти версии названы экспресс-алгоритмами. В связи с тем, что для своей реализации они предполагают полное распараллеливание соответствующих последовательных алгоритмов, предложены оценки выигрыша в быстродействии экспресс-алгоритмов по сравнению с классическими. Оценки показывают, что выигрыш может иметь порядок не меньший, чем $O(n)$. В перспективе предполагается разработать алгоритмы для коррекции экстремальных подграфов в динамических ГИС.

Библиографический список

1. Непомнящая А.Ш. Сравнение алгоритмов Прима-Дейкстры и Краскала с помощью ассоциативного параллельного процессора / А.Ш. Непомнящая // Кибернетика и системный анализ. — 2002. — № 2. — С. 19–28.
2. Рейнгольд Э. Комбинаторные алгоритмы. Теория и практика / Э. Рейнгольд, Ю. Нивергейт, Н. Део. — М.: Мир. 1980. — 476 с.
3. Ларионов А.М. Вычислительные комплексы, системы и сети / А.М. Ларионов, С.А. Майоров, Г. И. Новиков. — Л.: Энергоатомиздат, 1987. — 288 с.
4. Абрамский М.И. Система оценки эффективности параллельных алгоритмов / М.И. Абрамский, В.В. Кирюхин // Вестн. СевГТУ. — Севастополь, 2001. — Вып. №2. — С. 74–75.
5. Кирюхин В.В. Применение графовых моделей в работе геоинформационных систем / В.В. Кирюхин, М.И. Абрамский // Материалы междунар. науч.-практ. конф. «Инфотех-2002». — Севастополь, 2002. — С. 30–32.

Поступила в редакцию 12.03.2003 г.