УДК 519.688

ПАРАЛЛЕЛЬНЫЕ ЭВРИСТИЧЕСКИЕ АЛГОРИТМЫ ДЛЯ ЗАДАЧИ ВЕРШИННОЙ МИНИМИЗАЦИИ НЕДЕТЕРМИНИРОВАННЫХ КОНЕЧНЫХ АВТОМАТОВ

А. В. Цыганов, О. И. Булычов

Тольяттинский государственный университет

Поступила в редакцию 06.04.2010 г.

Аннотация. В статье рассматриваются параллельные версии генетического алгоритма и метода имитационной нормализации для задачи вершинной минимизации недетерминированных конечных автоматов. Предложенные алгоритмы реализованы в виде проекта с использованием библиотеки параллельных метаэвристик HeO.
Ключевые слова: Конечные автоматы, недетерминированные автоматы, вершинная минимизация, параллелизм, эвристики, метаэвристики.

Abstract. In the present paper, we consider parallel versions of the genetic algorithm and the simulated annealing method for the state minimization of nondeterministic finite automata. The proposed algorithms are implemented as a project that uses the library of parallel metaheuristics HeO.
Keywords: Finite automata, nondeterministic automata, state minimization, parallelism, heuristics, metaheuristics.

ВВЕДЕНИЕ

Конечные автоматы (КА) находят широкое применение в системах обработки информации. В качестве наиболее показательных примеров можно привести системы обработки больших массивов текста, проверки орфографии, распознавания речи и др.

Во многих приложениях важную роль играет способ представления КА в памяти компьютера. Прежде всего, перед разработчиками встаёт вопрос о выборе между двумя эквивалентными (с точки зрения описываемых ими языков) типами автоматов: недетерминированными конечными автоматами (НКА) и детерминированными конечными автоматами (ДКА).

В процессе работы с обоими типами КА может потребоваться их минимизация, которая может выполняться по разным критериям, например, по числу состояний (вершин), по числу переходов и др. Для ДКА минимизация числа состояний может быть выполнена за полиномиальное время, в то время как для НКА аналогичная задача является NP-трудной. Поэтому актуальным остаётся вопрос разработки эффективных приближённых методов решения задачи вершинной минимизации НКА.

1. ПОСТАНОВКА ЗАДАЧИ

Задача вершинной минимизации НКА состоит в построении такого автомата, который описывал бы тот же самый регулярный язык, что и исходный автомат, но имел бы при этом меньшее число состояний. Один из возможных подходов к вершинной минимизации НКА состоит в следующем. По двум КА, ассоциированным с исходным НКА, строится специальная таблица (матрица RAM в [1], отношение # в [2]), для которой минимизируется число прямоугольных блоков, содержащих все ее непустые элементы, затем по блокам, входящим в найденное покрытие, строится НКА. Если язык построенного автомата совпадает с языком исходного НКА, то искомый автомат построен, иначе ищется другое минимальное покрытие и т.д. Важнейшей и наиболее трудоёмкой подзадачей в данных алгоритмах (даже в случае относительно небольших НКА) является следующая. Задана прямоугольная матрица, заполненная элементами 0 или 1. Некоторую пару подмножеств строк и столбцов назовём блоком (гридом), если, во-первых, на всех их пересечениях стоят 1, и, во-вторых, это множество нельзя пополнить ни строкой, ни столбцом без нарушения первого свойства. Допустимым решением является множество блоков, покрывающих

все элементы 1 заданной матрицы. Требуется выбрать допустимое решение, содержащее минимальное число блоков — так называемое оптимальное решение.

На рис. 1 приведен пример матрицы, в которой имеются 5 блоков: $\alpha = \{A,B,C,D\} \times \{U\}$, $\beta = \{A,C,D\} \times \{Z,U\}$, $\gamma = \{B,C,D\} \times \{X,U\}$, $\delta = \{C,D\} \times \{X,Z,U\}$ и $\omega = \{D\} \times \{X,Y,Z,U\}$. Для покрытия всех значений 1 данной матрицы достаточно использовать 3 из этих 5 блоков: β, γ, и ω.

	X	Y	Z	U
A	0	0	1	1
B	1	0	0	1
C	1	0	1	1
D	1	1	1	1

Рис. 1. Пример матрицы с 5 блоками

Очевидным методом решения данной задачи является метод «грубой силы» (полный перебор), однако уже в случае нескольких десятков блоков данный метод становится очень затратным с вычислительной точки зрения. Одним из методов решения данной проблемы является использование эвристических (метаэвристических) алгоритмов, которые позволяют за приемлемое время получать решения близкие к оптимальным. В данной работе для решения рассматриваемой задачи предлагается использование параллельных версий метода имитационной нормализации (имитации отжига) и генетического алгоритма.

2. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Программная реализация предлагаемых методов была выполнена на языке C++ в виде проекта GridMin, использующего библиотеку метаэвристических алгоритмов оптимизации HeO (Heuristic Optimization). Описание первой версии данной библиотеки приведено в [3]. Основная идея данной библиотеки состоит в разделении программного кода алгоритма оптимизации на две части: проблемно-независимую и проблемно-зависимую. Первая часть реализована в библиотеке, а вторая реализуется пользователем. Алгоритмы оптимизации выполнены в виде так называемых алгоритмических каркасов, что позволяет применять их к широкому кругу оптимизационных задач и скрывать от пользователя все аспекты параллельной реализации. Использованные при реализации библиотеки шаблоны проектирования позволяют получать гибкие алгоритмы с возможностью гибридизации.

В настоящее время в библиотеке реализованы три алгоритма: генетический алгоритм (GA — Genetic Algorithm), метод имитационной нормализации или имитации отжига (SA — Simulated Annealing), а также их гибрид (GA+SA). Подробное описание последовательных версий методов GA и SA можно найти, например, в [4]. Каждый алгоритм библиотеки реализован с помощью двух технологий параллельного программирования: OpenMP и MPI. Основной моделью параллельных вычислений в библиотеке является мультистартовая модель: каждый из потоков выполняет последовательный код алгоритма и периодически происходит обмен решениями (кооперация). В библиотеке реализованы два режима кооперации: асинхронный и синхронный, схемы которых приведены на рис. 2. При синхронной кооперации осуществляется периодическое (через фиксированное число итераций) блокирование вы-

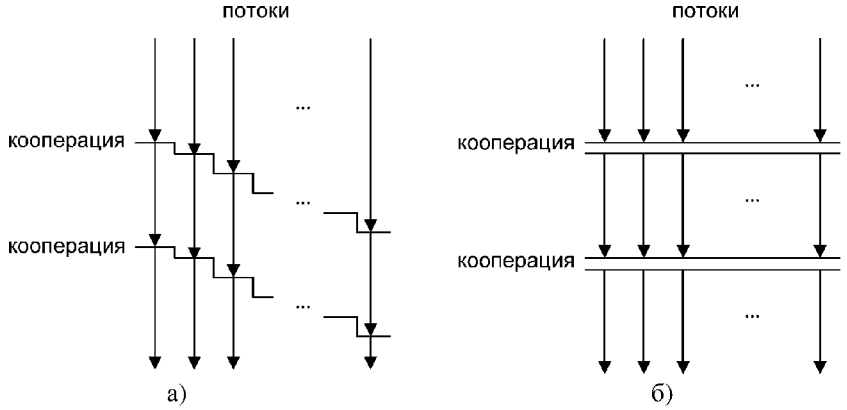

Рис. 2. Асинхронная (слева) и синхронная (справа) кооперации потоков

полнения потоков и обмен данными между ними. Асинхронная кооперация инициируется первым потоком через фиксированные промежутки времени и выполняется без блокировки потоков по мере их готовности к обмену данными. Использование синхронной кооперации может приводить к большой нагрузке на сеть в моменты кооперации. Асинхронный режим позволяет эффективно решать задачу на неоднородных кластерах и снизить нагрузку на сеть.

Использование мультистартовой модели позволяет с увеличением числа потоков увеличивать точность получаемого решения (ускорять сходимость). Для генетического алгоритма с использованием технологии OpenMP реализована также модель глобальной популяции, позволяющая с ростом числа потоков ускорять процесс вычислений.

Для реализации алгоритма поиска оптимального покрытия с использованием методов библиотеки HeO были реализованы проблемно-зависимые классы и специфичные для них методы. Для метода SA таковым является метод move() (очередной случайный ход), а для GA — методы cross() и mutate() (кроссовер и мутация). Основными проблемно-зависимыми классами проекта являются классы MinGrid_Problem и MinGrid_Solution, описывающие соответственно специфику решаемой задачи и ее допустимых решений. Основными структурами данных класса MinGrid_Problem являются матрица rel_ из нулей и единиц, и вектор grids_, в котором содержится описание каждого блока (грида) этой матрицы (для матрицы, изображенной на рис. 1, его размерность будет равна 5). Основной структурой данных класса MinGrid_Solution является двоичный вектор grids_set_ (каждый элемент этого вектора равен 1, если соответствующий блок из вектора grids_ включается в покрытие, и 0 в противном случае). Таким образом, потенциальные решения для обоих методов кодируются с помощью двоичных векторов, что позволяет реализовать для них методы move(), cross() и mutate() «стандартным образом» (например, в данном проекте для генетического алгоритма был реализован стандартный одноточечный кроссовер). Единственной проблемой является потенциальная недопустимость получаемых данными методами решений (т.е. наборы блоков, покрывающих не все 1 в матрице rel_). Данная проблема была решена следующим образом: в класс MinGrid_Solution была добавлена целочисленная матрица hits_, имеющая такой же размер, что и матрица rel_. Для каждого ненулевого элемента матрицы rel_ в матрице hits_ хранится число покрывающих его блоков. При добавлении или удалении блока из вектора grids_set_, значения матрицы hits_ пересчитываются. В начальных

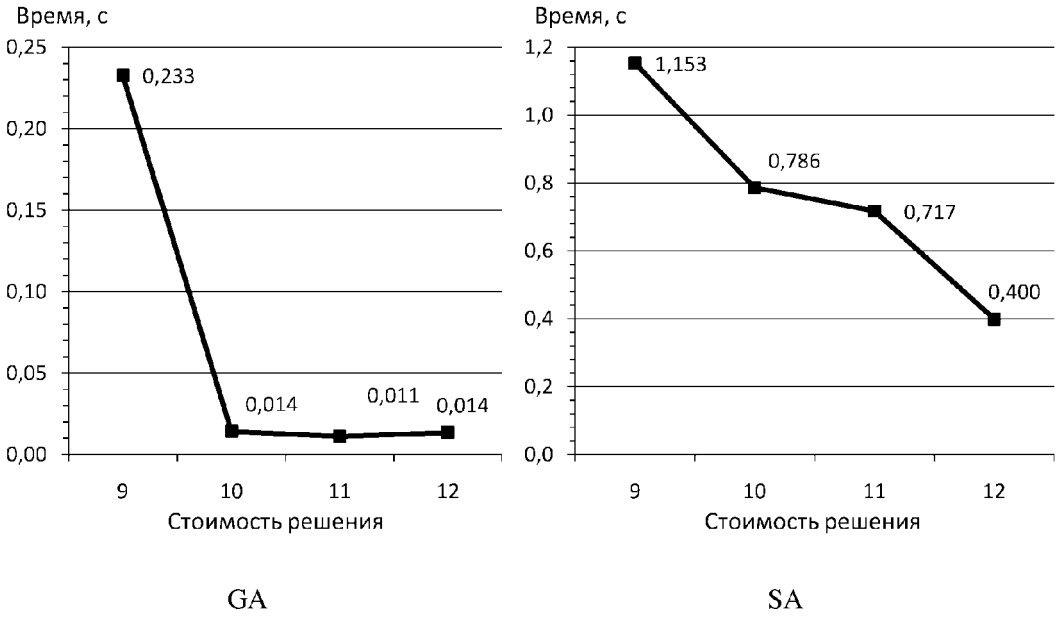

Рис. 3. Пример матрицы с 40 блоками и найденного для нее покрытия

решениях для обоих методов вектора grids_set_ состоят из одних единиц (тривиальные покрытия). Если при удалении некоторого блока из текущего покрытия счетчик числа покрывающих блоков в матрице hits_ для какой-либо 1 матрицы rel_ обнуляется, то такое покрытие признается недопустимым и блок возвращается в покрытие.

На рис. 3. приведены фрагменты отчета работы алгоритма SA для матрицы размера 10x10, содержащей 70 блоков. Время поиска решения зависит от настроек алгоритмов. На компьютере Intel Core 2 Quad Q6600 @ 2.4 ГГц время получения решения для данной задачи в 4 потока составило около 6 с. Заметим, что в найденном решении число блоков в покрытии оказалось меньше минимума из числа строк и столбцов, равного 10 (решения, не удовлетворяющие такому условию, являются в методе минимизации НКА заведомо неприемлемыми). Теперь по найденному покрытию после применения специальной процедуры может быть получен НКА с числом состояний меньше, чем у исходного.

ЗАКЛЮЧЕНИЕ

В данной работе были описаны параллельные эвристические алгоритмы для решения важной подзадачи вершинной минимизации недетерминированных конечных автоматов. Следует еще раз отметить, что предложенные методы являются приближенными. Их применение не гарантирует получение оптимальных решений. Кроме того, заметим, что выделение всех блоков матрицы является отдельной трудоемкой задачей.

В настоящее время авторами данной статьи ведется разработка программного обеспечения для точной и приближенной минимизации НКА с использованием методов, предложенных в [1] и [2].

СПИСОК ЛИТЕРАТУРЫ

1. *Kameda T.* On the state minimization of nondeterministic finite automata / T. Kameda, P. Weiner // IEEE Transactions on Computers. — 1970. — Vol. C—19, no. 7. — Pp. 617—627.

2. *Мельников Б. Ф.* Недетерминированные конечные автоматы. — Тольятти: Изд-во ТГУ, 2009. — 160 с.

3. *Цыганов А. В.* HeO: библиотека метаэвристик для задач дискретной оптимизации / А. В. Цыганов, О. И. Булычов // Программные продукты и системы. — 2009. — № 4. — С. 148—151.

4. *Громкович Ю.* Теоретическая информатика. Введение в теорию автоматов, теорию вычислимости, теорию сложности, теорию алгоритмов, рандомизацию, теорию связи и криптографию / Ю. Громкович. — 3-е изд. — СПб: БХВ-Петербург, 2009. — 336 с.

Цыганов Андрей Владимирович — кандидат физ.-мат. наук, старший научный сотрудник, Тольяттинский государственный университет. Тел. (903) 320-46-79, e-mail: andrew.tsyganov@gmail.com.

Булычов Олег Иванович — ведущий программист компании «Gladiators Software», Ульяновск. Тел. (951) 097-04-61, e-mail: oleg.bulychov@gmail.com.

Tsyganov A. V. — Candidat of Physics-math. Sciences, senior researcher, Togliatti State University. Tel. (903) 320-46-79. E-mail: andrew.tsyganov@gmail.com.

Bulychov O. I. — senior programmer, Gladiators Software Co., Ulyanovsk. Tel. (951) 097-04-61. E-mail: oleg.bulychov@gmail.com.