

УДК 681.3

Э.И. Ватутин, канд. техн. наук, доцент, кафедра вычислительной техники, ЮЗГУ (e-mail: evatutin@rambler.ru)

## **АНАЛИЗ ЭФФЕКТИВНОСТИ И ПРОГРАММНАЯ ОПТИМИЗАЦИЯ МЕТОДОВ СИНТЕЗА РАЗБИЕНИЙ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ В СРЕДЕ РАЕ**

*Приведены результаты анализа эффективности программных реализаций эвристических методов синтеза разбиений параллельных алгоритмов логического управления, работающих в составе программной системы РАЕ. На основании проведенного анализа осуществлена программная оптимизация, в результате которой время синтеза разбиений граф-схем алгоритмов из 100 вершин сокращается в 3,2 раза при использовании метода параллельно-последовательной декомпозиции и в 2,7 раза при использовании метода С.И. Баранова.*

*Ключевые слова: система логического управления, проектирование логических мультиконтроллеров, разбиения, параллельный алгоритм, дискретная комбинаторная оптимизация, оптимизация программных средств*

\*\*\*

Одним из перспективных подходов к созданию систем логического управления (СЛУ) является их реализация в базисе логических мультиконтроллеров [1]. При этом СЛУ представляет собой однородную структуру из группы взаимосвязанных логических контроллеров, которые в совокупности способны реализовывать произвольные параллельные алгоритмы логического управления теоретически неограниченной сложности. Ключевую роль при проектировании подобных мультисистем играет выбор разбиения заданной граф-схемы алгоритма, т.к. его числовые параметры (сложность сети межблочных связей, степени дублирования микроопераций и логических условий и др.) напрямую влияют на аппаратную сложность и быстродействие СЛУ. Задача выбора оптимального разбиения является задачей дискретной комбинаторной оптимизации и относится к классу вычислительно сложных, что не допускает отыскания оптимального решения для алгоритмов управления с большим числом вершин за приемлемое для практики время. Для ее решения на практике разработан ряд эвристических методов [2–5] и визуальная среда РАЕ для автоматического синтеза разбиений [6–7]. Вычислительные эксперименты, проводимые с использованием разработанных программных средств с целью сравнения методов в различных условиях, также являются достаточно затратными с точки зрения требований к необходимому объему вычислительного времени, поэтому вызывает интерес анализ «узких мест» (англ. hotspots или bottlenecks), ограничивающих их быстродействие, и их возможная программная оптимизация.

Одним из этапов, выполняемых непосредственно перед синтезом разбиений, является построение матрицы отношений вершин  $M_R = \|m_{ij}\|$ ,  $i, j = \overline{1, N}$  [8], где  $N$  – число вершин в составе граф-схемы алгоритма. Выяснение отношений следования  $\nu$  фактически сводится к нахождению его

транзитивного замыкания путем нахождения таких групп вершин  $a_i$ ,  $a_j$  и  $a_k$ , что  $(a_i \nu a_j) \wedge (a_j \nu a_k) \wedge \neg(a_i \nu a_k)$  до тех пор, пока это возможно. Для построения транзитивного замыкания может быть использован алгоритм Флойда-Уоршелла [9].

Его особенностью является порядок следования циклов, обеспечивающий нахождение искомого транзитивного замыкания за один проход выполнения вложенных циклов (т.е. за время  $O(n^3)$ ). На псевдокоде он может быть представлен в следующем виде:

```

for  $k := 1$  to  $n$  do
  for  $i := 1$  to  $n$  do
    if  $(a_i \nu a_k)$  then
      for  $j := 1$  to  $n$  do
        if  $(a_k \nu a_j)$  then
           $m_{ij} := \{ \nu \} ;$ 

```

При этом в качестве исходных значений для алгоритма выступает матрица отношения следования, в которой заполнены позиции для пар вершин, непосредственно соединенных дугой. В ранней реализации алгоритма выяснения отношения следования [3] порядок циклов был иным (первые два цикла поменяны местами):

```

for  $i := 1$  to  $n$  do
  for  $k := 1$  to  $n$  do
    if  $(a_i \nu a_k)$  then
      for  $j := 1$  to  $n$  do
        if  $(a_k \nu a_j)$  then
           $m_{ij} := \{ \nu \} ;$ 

```

Исследования на выборках алгоритмов управления со случайной структурой [10, 11] показывают, что использованный в ранней реализации порядок следования циклов приводит к построению матрицы отношения следования за два прохода (приведенные циклы выполняются дважды, до тех пор, пока матрица  $M_R$  перестает изменяться), в то время как использование порядка циклов, предложенного в алгоритме Флойда-Уоршелла, требует одного прохода.

Использование данной алгоритмической оптимизации при синтезе разбиений алгоритмов управления из 100 вершин параллельно-последовательным методом позволяет сократить общее время синтеза разбиения на 20,4% с 582 мс до 463 мс на процессоре Intel Core 2 Duo 1,86 ГГц (однопоточная реализация, число алгоритмов управления в выборке

$K = 1000$ , ограничения отсутствуют, длительность вычислительного эксперимента – 20 мин.).

В ранней программной реализации [3, 8] в ходе построения матрицы отношений вершин отношения выяснялись в следующем порядке: отношение следования  $\nu$ , отношение связи  $\varphi$ , отношение альтернативы  $\psi$ , отношение параллельности  $\omega$ . При этом выяснение отношения связи производилось непосредственно после отношения следования следующим образом:

```
for i := 1 to n do
  for j := 1 to n do
    if (a_i  $\nu$  a_j) then begin
      m_ij := m_ij  $\cup$  {  $\varphi$  };
      m_ji := m_ji  $\cup$  {  $\varphi$  };
    end;
```

Можно заметить, что выяснение отношения связи можно совместить с выяснением отношения следования. Для этого вместо присваивания

$$m_{ij} := \{ \nu \};$$

необходимо использовать пару присваиваний

$$\begin{aligned} m_{ij} &:= \{ \nu, \varphi \}; \\ m_{ji} &:= \{ \varphi \}; \end{aligned}$$

как на этапе инициализации отношения следования, так и в процессе его транзитивного замыкания, рассмотренном выше. Подобное совместное выяснение отношений следования и связи дополнительно снижает затраты времени на 3,9% с 463 мс до 445 мс.

В процессе синтеза разбиения на различных этапах работы методов производится формирование протокола выполняемых преобразований (в текстовом виде или в формате HTML) с последующим сохранением в файл. Подобный протокол может быть полезен в случае детального анализа качества полученного разбиения или выяснения причин его некорректности во время отладки/тестирования, однако при построении разбиений выбор алгоритмов управления в ходе вычислительных экспериментов на его формирование может быть затрачено достаточно большое количество времени, поэтому вызывает интерес добавление в среду возможности управления его формированием с использованием булевой переменной-флажка. В результате отключения возможности формирования протокола промежуточных преобразований время синтеза разбиения сокращается на 57,5% (в 2,4 раза) с 445 мс до 190 мс. Таким образом, большую часть времени занимает именно формирование протокола преобразований (фактически, работа со строками), а не выполнение самих преобразований.

Аналогичную оптимизацию можно проделать с программными реализациями всех остальных методов. Так, например, для метода С.И. Баранова [4] также наблюдается сокращение времени синтеза разбиения на 62,3% (в 2,7 раза) с 15,1 мс до 5,7 мс (при тех же условиях проведения эксперимента).

В итоге указанные оптимизации приводят к уменьшению интегрального времени выполнения вычислительных экспериментов. Например, длительность вычислительного эксперимента с участием методов С.И. Баранова, А.Д. Закревского и параллельно-последовательного сокращается на 23,6% с 17,8 мин до 13,6 мин.

В процессе разработки среды РАЕ [6, 7] первоначально был реализован параллельно-последовательный метод (включающий в своем составе программную реализацию алгоритма построения матрицы отношений), который обменивался с графической оболочкой через интерфейс SMI (получение исходных данных и параметров разбиения, возврат разбиения). Позднее с появлением в составе среды новых методов синтеза разбиений (например, метод С.И. Баранова) появилась необходимость в реализации возможности их обращения к матрице отношений. С этой целью была реализована пара интерфейсов: RMBI (для построения матрицы отношений, на данный момент поддерживается только параллельно-последовательным методом) и RMUI (для использования уже построенной матрицы отношений, поддерживается всеми методами, требующими доступа к матрице отношений). Порядок действий при синтезе разбиения каждого алгоритма управления можно схематично представить следующим образом:

```
Построить_матрицу_отношений_для_RMBI( ) ;  
  
for i := 1 to n do  
    методi.Построить_разбиение( ) ;
```

При подобном подходе без ряда специальных проверок получается что построение матрицы отношений (а вместе с ней загрузка исходной граф-схемы алгоритма и ее предварительные преобразования) производится дважды для метода, реализующего интерфейс RMBI (первый раз при построении через RMBI, второй – при синтезе разбиения через SMI). Чтобы избежать этого в ходе оптимизации была предусмотрена возможность сохранения результатов предыдущих преобразований для метода, реализующего интерфейс RMBI, что сокращает время синтеза разбиения параллельно-последовательным методом на 3,7% с 190 мс до 183 мс.

Побочным эффектом подобной оптимизации является то, что время синтеза разбиения параллельно-последовательным методом теперь не включает затрат на построение матрицы отношений, что, с одной стороны, позволяет сравнивать все методы в равных условиях (например, временные затраты метода С.И. Баранова в работе [11] не учитывают построение матрицы отношений, хотя метод ее использует), однако, с другой стороны, несколько

затрудняет подсчет «честного» (с учетом построения матрицы отношений) времени получения разбиения в результате его прямого замера. Полученный малый выигрыш во времени свидетельствует о том, что построение матрицы отношений (вместе с подготовительными действиями) занимает достаточно малое время от общего времени построения разбиения параллельно-последовательным методом.

Таким образом, предварительный анализ затрат вычислительного времени на различные преобразования и проделанная программная оптимизация позволили сократить время синтеза разбиений с 582 мс до 183 мс (в 3,2 раза) для программной реализации параллельно-последовательного метода и с 15,1 мс до 5,7 мс (в 2,7 раза) для программной реализации метода С.И. Баранова, что открывает путь для более эффективного выполнения дальнейших вычислительных экспериментов.

### Список литературы

1. Ватутин Э.И. Проектирование логических мультиконтроллеров. Синтез разбиений параллельных граф-схем алгоритмов. Saarbrücken: Lambert Academic Publishing, 2011 г. 292 с.
2. Ватутин Э.И., Зотов И.В. Метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Параллельные вычисления и задачи управления (РАСО'04). М.: Институт проблем управления им. В.А. Трапезникова РАН, 2004. С. 884–917.
3. Ватутин Э.И., Зотов И.В. Параллельно-последовательный метод формирования субоптимальных разбиений параллельных управляющих алгоритмов // Свидетельство об официальной регистрации программы для ЭВМ № 2005613091 от 28.11.05.
4. Баранов С.И., Журавина Л.Н., Песчанский В.А. Метод представления параллельных граф-схем алгоритмов совокупностями последовательных граф-схем // Автоматика и вычислительная техника. 1984. № 5. С. 74–81.
5. Ватутин Э.И. Библиотека функций построения разбиений методом С.И. Баранова с жадным последовательным формированием блоков // Свидетельство о государственной регистрации программы для ЭВМ № 2010612902 от 28.04.10.
6. Ватутин Э.И., Зотов И.В. Программная система для построения разбиений параллельных управляющих алгоритмов // Идентификация систем и задачи управления (SICPRO'06). М.: Институт проблем управления им. В.А. Трапезникова РАН, 2006. С. 2239–2250.
7. Ватутин Э.И., Зотов И.В. Визуальная среда синтеза разбиений параллельных алгоритмов логического управления // Свидетельство об официальной регистрации программы для ЭВМ № 2007613222 от 30.07.07.

8. Ватутин Э.И., Зотов И.В. Построение матрицы отношений в задаче оптимального разбиения параллельных управляющих алгоритмов // Известия курского государственного технического университета. Курск, 2004. № 2. С. 85–89.
9. [http://ru.wikipedia.org/wiki/Алгоритм\\_Флойда\\_—\\_Уоршелла](http://ru.wikipedia.org/wiki/Алгоритм_Флойда_—_Уоршелла)
10. Vatutin E.I. Constructing Random Sample Parallel Logic Control Algorithms // 11th International Student Olympiad on Automatic Control (Baltic Olympiad BOAC'06). Saint-Petersburg, 2006. PP. 162–166.
11. Ватутин Э.И., Волобуев С.В., Зотов И.В. Комплексная сравнительная оценка методов выбора разбиений при проектировании логических мультиконтроллеров // Идентификация систем и задачи управления (SICPRO'08). М.: Институт проблем управления им. В.А. Трапезникова РАН, 2008. С. 1917–1940.