Многопроцессорные вычислительные системы
2006. №1(11)

УДК 681.5.015.63-192

© 2006 г. **Г.Б. Диго,**
Н.Б. Диго
(Институт автоматики и процессов управления ДВО РАН, Владивосток)

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА АППРОКСИМАЦИИ ОБЛАСТИ РАБОТОСПОСОБНОСТИ ВЫПУКЛЫМ МНОГОГРАННИКОМ[1]

Рассматривается задача аппроксимации выпуклым многогранником области работоспособности аналоговых технических устройств и систем при проектировании и управлении с учетом параметрических возмущений. Приводится параллельный аналог алгоритма поиска вершин многогранника и решения системы линейных неравенств. Проанализированы результаты программной реализации.

Введение

При построении областей работоспособности в задачах параметрического синтеза приходится иметь дело с решением систем алгебраических неравенств. Так, при аппроксимации их выпуклыми многогранниками [1, 2] исходят из математических моделей исследуемых систем, условий работоспособности, задаваемых полиномами не выше второго порядка, и ограничений на внутренние параметры. Используемый для этого алгоритм состоит из анализа ограничивающих область работоспособности неравенств на выпуклость, замены невыпуклых неравенств опорными гиперплоскостями (линейными неравенствами), наилучшим образом их приближающими, кусочно-линейного приближения каждого выпуклого неравенства несколькими линейными, решения полученной системы линейных неравенств, задающей искомый выпуклый многогранник, однозначно определяемый координатами своих вершин.

Ускорение процесса вычислений достигается путем их распараллеливания при проверке нелинейных неравенств на выпуклость, замене каждого выпуклого неравенства несколькими линейными неравенствами, поиске опорных гиперплоскостей для невыпуклых неравенств и решении системы линейных неравенств на этапе нахождения координат вершин вы-

[1] Работа выполнена при финансовой поддержке гранта РФФИ 05-08-01398 и гранта ДВО РАН 06-П15-056 по программе №15 ОЭММПУ РАН.

пуклого многогранника, аппроксимирующего область работоспособности.

Статья посвящена программной реализации параллельного алгоритма решения системы линейных неравенств на этапе поиска координат вершин ограничивающего область работоспособности выпуклого многогранника, предложенного в [3].

Постановка задачи

Пусть исследуется система, качество работы которой зависит от значений параметров ее элементов: $\mathbf{x} = (x^{(1)},...,x^{(n)})$, $\mathbf{x} \in R^n$,

$$x_{i\min} \le x_i \le x_{i\max},\ i = \overline{1,n}, \qquad (1)$$

а условия ее работоспособности заданы в виде системы неравенств

$$a_j \le y_j(\mathbf{x}) \le b_j, \quad j = 1,...,m, \qquad (2)$$

где $\mathbf{y} = \{y_j\}_{j=1}^m$ – вектор выходных параметров устройства, причем $y_j = F_j(x^{(1)},...,x^{(n)})$, а $F_j(\cdot)$ – известный оператор, зависящий от топологии исследуемого устройства.

Предположим, что $y_j(\mathbf{x})$, $j = 1,...,m$ из (2) являются полиномами не выше второго порядка и заданы ограничения на внутренние параметры $\mathbf{x} = (x^{(1)},...,x^{(n)})$, $\mathbf{x} \in R^n$. Тогда область работоспособности $D_\mathbf{x}$ можно представить системой в общем случае нелинейных неравенств [1]

$$B_j(x^{(1)}, x^{(2)},...,x^{(n)}) \le 0, \quad j = 1,2,...,2(m+n), \qquad (3)$$

в которой последние $2n$ неравенств являются линейными ограничениями (1) на внутренние параметры

$$B_{2m+j}(x^{(1)}, x^{(2)},...,x^{(n)}) = -x^{(j)} + x_{\min}^{(j)},$$

$$B_{2m+n+j}(x^{(1)}, x^{(2)},...,x^{(n)}) = x^{(j)} - x_{\max}^{(j)},$$

$$j = 1,\mathbf{K},n.$$

Ее существование следует из совместности (3). Ввиду отсутствия общих методов решения систем нелинейных неравенств был разработан алгоритм перехода к системе линейных неравенств, решаемой известными методами [2 – 4]. Используемый подход равносилен аппроксимации области, ограниченной системой нелинейных неравенств, выпуклым многогранником, описываемым системой линейных неравенств. Геометрически он представляет собой общую часть полупространств, ограниченную гиперплоскостями его граней, и однозначно определяется координатами своих вершин. Отсутствие вершин, удовлетворяющих исследуемой системе неравенств, означает ее несовместность и требует уточнения ограничений. Число возможных вершин искомого многогранника оценивается сверху величиной $g = C_{2(m+n)}^n$, где g – число сочетаний из $2(m+n)$ по n. С ростом размерности векторов внутренних и выходных параметров увеличивается число неравенств в системе (3), что существенно удлиняет процесс вычислений. Технология распараллеливания позволяет сократить временные затраты на этапе поиска координат вершин ограничивающего область работоспособности выпуклого многогранника.

Ставится задача оценки эффективности замены последовательного алгоритма на его параллельный аналог.

Анализ задачи

Очевидно, что система (3) может содержать нелинейные выпуклые и невыпуклые и линейные неравенства. Используемый в [2 – 4] алгоритм обеспечивает замену нелинейных неравенств линейными, преобразуя систему (3) в

$$\sum_{j=1}^{n} A_{ij} x^{(j)} - C_i \le 0, \quad i = 1, 2, \ldots, q, \tag{4}$$

где C_i – вободный член, $q = \sum_{p=1}^{k_1} i_p + k_2 + 2n$.

От системы (4) переходим к эквивалентной ей несократимой подсистеме независимых линейных неравенств

$$\sum_{j=1}^{n} T_{ij} x^{(j)} - a_i \le 0, \quad i = 1, 2, \ldots, l, \quad l \le q, \tag{5}$$

удаляя из нее зависимые неравенства.

В основу алгоритма получения системы (5), эквивалентной (4), положена вычислительная схема удаления зависимых неравенств из исходной системы [5]. Этот алгоритм не требует для реализации значительного времени и не обладает явным параллелизмом.

Система (5) позволяет находить координаты вершин соответствующего ей выпуклого многогранника, ограниченного гиперплоскостями его граней [6]

$$\sum_{j=1}^{n} T_{ij} x^{(j)} - a_i = 0, \quad i = 1, 2, \ldots, l, \quad l \le q. \tag{6}$$

Каждая вершина является пересечением n гиперплоскостей из (6), удовлетворяющим системе (5). Их общее число не превосходит C_l^n (число сочетаний из l по n) и не меньше 1 (совместная система (5) имеет хотя бы одну вершину).

Обозначив $\mathbf{T} = (T_{ij})$, $\mathbf{a} = (a_1, \ldots, a_l)$, представим систему (5) в матричной форме

$$\mathbf{Tx} - \mathbf{a} \le \mathbf{0}. \tag{7}$$

Для поиска координат вершин из матрицы $\mathbf{T}$ поочередно формируются подматрицы $\mathbf{T}_i$ $(i = 1,2,\ldots,C_l^n)$, содержащие те же столбцы, что и матрица $\mathbf{T}$, и n ее строк, выбираемых как i-е сочетание из l по n, начиная с первого: строки матрицы $\mathbf{T}$ с номерами $1,2,\ldots,n$.

Процедура поиска конкретной вершины заключается в вычислении и проверке на равенство нулю определителя $\det \mathbf{T}_i$ подматрицы $\mathbf{T}_i$, вычислении координат предполагаемой вершины при $\det \mathbf{T}_i \neq 0$ и запоминании их при удовлетворении системе (7).

Параллельный алгоритм решения системы линейных неравенств

Применительно к алгоритму, изложенному выше, технология распараллеливания заключается в следующем.

Корневой процессор рассылает матрицу $\mathbf{T}$ доступным k процессорам.

Из матрицы $\mathbf{T}$ на каждом j-м процессоре формируется подматрица $\mathbf{T}_j$, $j = 1,2,\ldots,k$, содержащая то же число столбцов, что и матрица $\mathbf{T}$, и n ее строк, номера которых совпадают с j-м сочетанием из l по n, начиная с первого: строки матрицы $\mathbf{T}$ с номерами $1,2,\ldots,n$.

Для каждой подматрицы $\mathbf{T}_j$ ищется предполагаемая вершина многогранника, координаты существующей вершины запоминаются.

Процесс формирования подматриц $\mathbf{T}_j$ и соответствующий поиск вершин на k процессорах повторяется до тех пор, пока не будут просмотрены все возможные сочетания.

Полученные результаты собираются на корневом процессоре.

Процедура поиска конкретной вершины многогранника на отдельном процессоре:

1) вычисление определителя Δ матрицы $\mathbf{T}_j$;

2) проверка на равенство нулю определителя Δ:

если $\Delta = 0$, вершины нет, переход к 4);

если $\Delta \neq 0$, вычисляются координаты предполагаемой вершины;

3) проверка принадлежности предполагаемой вершины многограннику:

если предполагаемая вершина удовлетворяет всем неравенствам системы (7), то она является вершиной многогранника, ее координаты и номера гиперплоскостей, ей соответствующих, запоминаются в массиве координат вершин и в массиве номеров гиперплоскостей;

если предполагаемая вершина не удовлетворяет хотя бы одному из неравенств системы (7), то переход к 4);

4) окончание поиска конкретной вершины.

Программная реализация

Для реализации выбрана модель программы SPMD (Simple Program-Multiple Data) – одна и та же программа работает на разных процессорах со своими данными, осуществляя распараллеливание по данным (одни и те же потоки команд управляют различными потоками данных). Структура обмена однородна в пределах одного блока и между блоками.

Среда параллельного программирования реализована на базе интерфейса передачи сообщений MPI (Message Passing Interface), обеспечивающего связь между ветвями параллельного приложения. Стандарт MPI использует следующую модель программирования: параллельное приложение включает несколько одновременно выполняемых процессов, обменивающихся между собой данными с помощью сообщений. Модель передачи сообщений является основной моделью параллельного выполнения программы, представляющей собой систему процессов, взаимодействующих посредством передачи сообщений. MPI-программа содержит набор независимых процессов, выполняющих свою собственную программу (не обязательно одну и ту же), написанную (в данном случае) на языке фортран77. Для работы с MPI используется реализация LAM (Local Area Multicomputer-локальный мультикомпьютер), версия 7.1.1/MPI, установленная на многопроцессорной системе МВС-1000/16 в ИАПУ ДВО РАН.

LAM-приложение инициируется из командной строки командой `lamboot`. Для компиляции и сборки программного модуля используется команда mpif77 с соответствующими параметрами. В LAM-приложении SPMD-программы выполняются с помощью схемы приложения (application schema), определяющей число процессов и вычислительные ресурсы. Запуск программы осуществляется командой `mpirun` со схемой приложения в качестве аргумента либо этой же командой с заданием числа процессоров и имени исполняемого файла в качестве параметров.

Алгоритм реализован на МВС-1000/16 под управлением ОС LINUX. Разработанная программа протестирована на системах линейных неравенств с разным количеством неравенств и числом переменных. Ниже приведены результаты, полученные для системы из 14 неравенств с 8 параметрами.

Тестовый пример

Начальные данные.

Матрица **T** из (7):

-0.4025 –0.4025 –0.4025 –0.4025 0. 0. 0. 0.
-0.54304 –0.14054 0.26196 0.66446 0. 0. 0. 0.
-0.51326 0.8539 0.61106 –1.2418 0. 0. 0. 0.
0.2257 0.14054 –0.2257 –0.06802 0. 0. 0. 0.
0.96466 –0.5003 –0.15966 1.0152 0. 0. 0. 0.
0.09164 –0.14054 0.18944 –0.23834 0. 0. 0. 0.

-1. 0. 1. 2. 0. 0. 0. 0.
0. 0. 0. –1. 0. 1. 2.
-0.021869 –0.021869 –0.021869 –0.021869 1. 1. 1. 1.
-0.021869 –0.021869 –0.021869 –0.021869 –1. –1. –1. –1.
3.1652 3.1652 3.1652 3.1652 –1. –1. –1. –1.
-3.4742 –3.4742 –3.4742 –3.4742 1. 1. 1. 1.

Вектор **a**:

-0.1E-07 0.4025 5.7902 4.0639 22.285 7.4697 –0.1E-07 –0.1E-07 1. 1. 1. 1.

Результаты расчета.

Возможное число вершин IC= 495.

Случай двух процессоров

```
N=2

nc= 247 nrest= 1
 Process  0 of  2 started for array from  1 till  248
  time calculation=   0.653778076 myid= 0
nc= 247 nrest= 1
 Process  1 of  2 started for array from  249 till  495
  time calculation=   0.67544198 myid= 1
```

Случай трех процессоров

```
N=3

nc= 165 nrest= 0
 Process  0 of  3 started for array from  1 till  165
  time calculation=   0.669680119 myid= 0
nc= 165 nrest= 0
 Process  1 of  3 started for array from  166 till  330
  time calculation=   0.678686142 myid= 1
nc= 165 nrest= 0
 Process  2 of  3 started for array from  331 till  495
  time calculation=   0.672533989 myid= 2
```

Случай четырех процессоров

```
N=4

nc= 123 nrest= 3
 Process  0 of  4 started for array from  1 till  124
  time calculation=   0.780846119 myid= 0
nc= 123 nrest= 3
 Process  1 of  4 started for array from  125 till  248
  time calculation=   0.816464186 myid= 1
nc= 123 nrest= 3
 Process  2 of  4 started for array from  249 till  372
  time calculation=   0.813391924 myid= 2
nc= 123 nrest= 3
 Process  3 of  4 started for array from  373 till  495
  time calculation=   0.818017006 myid= 3
```

Случай пяти процессоров

```
N=5

nc= 99 nrest= 0
 Process  0 of  5 started for array from  1 till  99
  time calculation=   0.684187174 myid= 0
nc= 99 nrest= 0
 Process  2 of  5 started for array from  199 till  297
  time calculation=   0.693640947 myid= 2
```

```
nc= 99 nrest= 0
 Process  4 of  5 started for array from  397 till  495
  time calculation=   0.68405509 myid= 4
nc= 99 nrest= 0
 Process  3 of  5 started for array from  298 till  396
  time calculation=   0.699374199 myid= 3
nc= 99 nrest= 0
 Process  1 of  5 started for array from  100 till  198
  time calculation=   0.725219011 myid= 1
```

Случай шести процессоров

```
N=6

nc= 82 nrest= 3
 Process  0 of  6 started for array from  1 till  83
  time calculation=   0.778204203 myid= 0
nc= 82 nrest= 3
 Process  2 of  6 started for array from  167 till  249
  time calculation=   0.791912079 myid= 2
nc= 82 nrest= 3
 Process  4 of  6 started for array from  332 till  413
  time calculation=   0.782993078 myid= 4
nc= 82 nrest= 3
 Process  1 of  6 started for array from  84 till  166
  time calculation=   0.817204952 myid= 1
nc= 82 nrest= 3
 Process  3 of  6 started for array from  250 till  331
  time calculation=   0.812289953 myid= 3
nc= 82 nrest= 3
 Process  5 of  6 started for array from  414 till  495
  time calculation=   0.788336039 myid= 5
```

Количество вершин аппроксимирующего многогранника

```
  IVM=  44 K=  8
```

Заключение

При аппроксимации области работоспособности выпуклым многогранником предложены параллельный алгоритм на этапе поиска его вершин и программа, реализующая этот алгоритм решения системы линейных неравенств. Для программы выбрана модель SPMD. Результаты тестирования на конкретных данных подтвердили ускорение вычислений при использовании технологии распараллеливания. Время вычислений варьировало в зависимости от числа доступных процессоров.

ЛИТЕРАТУРА

1. *Абрамов О.В., Диго Г.Б., Диго Н.Б., Катуева Я.В.* Параллельные алгоритмы построения области работоспособности // Информатика и системы управления. – 2004. – № 2(8). – С. 121-133.
2. *Бернацкий Ф.И., Диго Г.Б., Диго Н.Б.* Распараллеливание вычислений при построении областей допустимых управлений // Тр. междунар. симпозиума «Надежность и качество». – Пенза: ПГУ. 2003. – С. 24-26.
3. *Бернацкий Ф.И., Диго Г.Б., Диго Н.Б.* Параллельный алгоритм решения системы линейных неравенств для построения областей допустимых управлений // Тр. междунар. симпозиума «Надежность и качество». – Пенза: ПГУ. 2003. – С. 20-21.

4. *Бернацкий Ф.И., Диго Г.Б., Диго Н.Б.* Построение области робастных управлений на параллельных процессорах // Информатика и системы управления. – 2003. – № 1(5). – С. 92-100.
5. *Черников С.Н.* Линейные неравенства. – М.: Наука, 1968.
6. *Александров А.Д.* Выпуклые многогранники. – М.-Л.: ГИТТЛ, 1950.

Статья представлена к публикации членом редколлегии О.В. Абрамовым.

УДК 681.5.015.63-192

© 2006 г. **А.А. Евдокименко,**
Я.В. Катуева, канд. техн. наук
(Институт автоматики и процессов управления ДВО РАН, Владивосток)

НАХОЖДЕНИЕ ЦЕНТРА ТЯЖЕСТИ ОБЛАСТИ РАБОТОСПОСОБНОСТИ, ЗАДАННОЙ В МАТРИЧНОЙ ФОРМЕ[1]

Рассматривается задача нахождения центра тяжести областей работоспособности, возникающая в процессе проектирования и управления с учетом параметрических возмущений. Для ее решения разработаны алгоритмические и программные средства, в основе которых лежит идея распараллеливания процессов поиска конечного результата.

Введение

Одним из этапов процесса проектирования технических устройств и систем с учетом возможных вариаций параметров их элементов является нахождение множества допустимых изменений параметров (области работоспособности)[1]. Знание области работоспособности исследуемого устройства необходимо для решения ряда различных задач, обычно рассматриваемых в теории надежности. Среди них можно выделить следующие задачи:

оценка способности системы сохранять работоспособность при отклонениях параметров ее элементов от расчетных значений;

оценка чувствительности параметров и выделение определяющих входных параметров;

назначение допусков на параметры;

[1] Работа выполнена при финансовой поддержке гранта РФФИ 05-08-01398 и гранта ДВО РАН 06-3-03 «Разработка комплекса алгоритмов и программ параметрического синтеза по критерию надежности».