



МЕТОДЫ ДЕКОМПОЗИЦИИ МОДЕЛЕЙ НЕПРЕРЫВНЫХ СИСТЕМ ДЛЯ МОДЕЛИРОВАНИЯ В УСЛОВИЯХ РАСПРЕДЕЛЕННЫХ РЕСУРСОВ

ГОРБАЧЕВ В.А., ВОЛК М.А., БАБАЕВ А.П.

Возрастающая сложность современных систем и высокие требования к срокам создаваемых проектов определяют необходимость использования быстродействующих методов и алгоритмов автоматизированного проектирования. Приведенные в статье методы декомпозиции и компоновки моделей непрерывных систем могут быть использованы при разработке алгоритмов параллельного моделирования систем в условиях распределенных вычислительных ресурсов.

Широкое распространение и доступность ПЭВМ сделало ее одними из наиболее используемых инструментов для проведения научных исследований. Сегодня одним из свойств персональных компьютеров стало их включение в многомашинные комплексы, локальные и глобальные сети. В последние два-три года наблюдается тенденция к использованию прикладными и операционными программами вычислительных ресурсов всех компьютеров (процессоров), доступных пользователю.

Далее рассматриваются традиционные методы повышения быстродействия процесса моделирования непрерывных систем с точки зрения их использования для выполнения параллельных вычислений на уровне подсистем. С этой целью вводится понятие декомпозиции системы и приводятся некоторые эвристические правила ее проведения. Учитывая специфику вычислительных задач, возникающих в рамках данного процесса, описанный в работе метод подсхем ориентирован на реализацию численных методов решения систем уравнений, являющихся моделью системы. Рассмотрим один из возможных алгоритмов компоновки модели, проведем его анализ и возможные модификации.

1. Декомпозиция моделей систем

Согласно общей теории систем модель системы можно представить в виде двойки

$$M = [A, S], \quad (1)$$

где A — непустое множество элементов; S — множество связей между элементами.

Введем понятие частной модели системы M_i , которую определим как

$$M_i = [A_i, S_i], A_i \subset A, S_i \subset S. \quad (2)$$

Под декомпозицией модели будем понимать такое множество частных моделей M_i , $i = \overline{1, N}$, где

$$M = \bigcup_{i=1}^N M_i = \bigcup_{i=1}^N [A_i, S_i] \quad (3)$$

(N — число частных моделей).

Введем условия проведения декомпозиции:

1. Множества элементов частных моделей не должны пересекаться:

$$A_i \cap A_j = \emptyset, i, j = \overline{1, N}, i \neq j.$$

2. Связи между элементами могут принадлежать нескольким частным моделям; в этом случае элементы, объединенные такой связью, должны принадлежать разным частным моделям.

3. Вычислительная стоимость обработки частных моделей должна быть приблизительно одинакова. Под вычислительной стоимостью будем понимать показатель, оценивающий время и ресурсы, требуемые для моделирования частной модели. Если элементы и связи системы однородны или их можно в приближении представить таковыми, то при разбиении модели можно пользоваться следующим принципом: мощность частных моделей M_i выбирается приблизительно равной, а количество взаимосвязей между частными моделями должно быть минимально. Число элементов в частной модели можно оценить как

$$\text{count}(A_i) \approx \frac{\text{count}(A)}{N}.$$

4. Количество частных моделей N выбирается с учетом доступных вычислительных ресурсов и скорости каналов обмена данными.

5. Условие информативности декомпозиции заключается в объединении в одну частную модель элементов, относящихся к одной подсистеме. В качестве критерия при формировании множества элементов частной модели могут выступать, например, показатели более “сильных” связей. Соблюдение этого принципа также позволяет организовать изучение самой подсистемы.

Следует отметить, что большинство из перечисленных условий на сегодняшний день нельзя реализовать формально, и при построении алгоритмов декомпозиции чаще всего данные условия реализуются в эвристических правилах. На практике, при декомпозиции конкретной модели для конкретных целей пользуются одним (несколькими) критерием эффективности.

Выше изложен общий принцип декомпозиции систем. Далее рассмотрим его применение для моделей, отражающих непрерывный объект, с учетом определенных особенностей построения аналоговой модели, которые позволяют значительно поднять эффективность моделирования. Приведенные ниже методы и алгоритмы были исследованы при построении системы моделирования аналоговых электронных схем. Без потерь в общности описания далее будем использовать наряду с термином “система” понятие “схема”.

2. Метод подсхем

Рассмотрим идею метода подсхем [1]. Пусть имеется схема G . Ее моделью является система:

$$W(x) = 0. \quad (4)$$

Здесь x – вектор переменных схемы (в качестве которых выступают узловые потенциалы, токи).

Разобьем схему G на N подсхем g_1, g_2, \dots, g_N . Уравнения каждой из них можно записать в виде

$$Wg_1(x_\Gamma, x_{B1})=0, Wg_2(x_\Gamma, x_{B2})=0, \dots, Wg_N(x_\Gamma, x_{BN})=0, \quad (5)$$

где x_Γ – граничные переменные подсхем, например, потенциалы внешних узлов или напряжения на внешних ветвях подсхемы, образующих с другими подсхемами; x_{B1}, \dots, x_{BN} – внутренние переменные подсхем.

Уравнение всей схемы можно записать в виде

$$W_\Gamma(x_\Gamma, x_{B1}, \dots, x_{BN}) = 0. \quad (6)$$

Решая выражения подсхем (5) относительно x_{B1}, \dots, x_{BN} , получаем

$$x_{B1} = Fg_1(x_\Gamma), x_{B2} = Fg_2(x_\Gamma), \dots, x_{BN} = Fg_N(x_\Gamma). \quad (7)$$

Эти соотношения отражают связи внутренних и граничных переменных каждой подсхемы. Подставляя их в (5), получаем уравнение схемы с исключенными внутренними переменными подсхем; их можно назвать равенствами сокращенной схемы:

$$W_\Gamma(x_\Gamma, Fg_1(x_\Gamma), Fg_2(x_\Gamma), \dots, Fg_N(x_\Gamma)) = W_\Gamma(x_\Gamma) = 0. \quad (8)$$

Размер этой системы определяется только числом граничных переменных, которое может быть много меньше общего числа переменных $x_\Gamma, x_{B1}, \dots, x_{BN}$. Решая (8), находим x_Γ . Подставляя x_Γ в уравнение связи (7), определяем x_{B1}, \dots, x_{BN} .

Таким образом, вместо решения системы уравнений полной схемы размера $[x_\Gamma] + [x_{B1}] + \dots + [x_{BN}]$, где $[x]$ – размер вектора x , здесь раздельно решаются $N+1$ системы равенств меньшего размера.

3. Применение декомпозиции при реализации численных методов на ЭВМ

Модель непрерывной системы в общем случае – это система дифференциальных нелинейных уравнений. Процесс моделирования представляет собой многократное решение этой системы, отражающее поведение системы во времени [1]. Применение численных методов, а именно они используются при моделировании на ЭВМ, приводит к тому, что основной вычислительной задачей на каждой итерации обычно является система линейных уравнений

$$Ax = B. \quad (9)$$

Рассмотрим применение идей декомпозиции для задачи распараллеливания процесса формирования и решения систем линейных уравнений. Согласно методу подсхем исходная схема разбивается на части – подсхемы или многополюсники, описываемые уравнениями вида

$$A_i x_i = B_i, \quad (10)$$

где A_i, B_i – матрица коэффициентов и вектор свободных членов i -й подсхемы; $i = \overline{1, N}$; $x_i = (x_{i\Gamma}, x_{iB})$ – вектор, включающий граничные $x_{i\Gamma}$ и внутренние x_{iB} переменные i -й подсхемы.

Разбиение на подсхемы можно производить разными способами [1]: методом сечений узлов, при котором узел может принадлежать одной или нескольким подсхемам (т. е. быть либо внутренними, либо граничными), а элемент принадлежит только одной подсхеме; методом контуров, при котором и

узел, и элемент могут принадлежать одной или нескольким подсхемам; методом сечений ветвей, при котором узел принадлежит одной подсхеме, а элемент – одной или нескольким.

Одновременно с разбиением формируется топологическое множество D_Γ типа узел-подсхема, контур-подсхема или сечение-подсхема в зависимости от типа переменных $x_{i\Gamma}$, указывающее, какие подсхемы участвуют при суммировании в граничных узлах, сечениях или в граничных контурах.

Уравнение (10) путем исключения внутренних переменных приводится к виду

$$A_{i\Gamma} x_{i\Gamma} = B_{i\Gamma}. \quad (11)$$

При этом размер системы (10) понижается на число внутренних переменных. Одновременно с (11) формируются уравнения связи граничных и внутренних переменных каждой подсхемы

$$x_{iB} = Fg_i(x_{i\Gamma}). \quad (12)$$

В нашем случае равенство (10) линейное, поэтому формирование (11) и (12) выполняется следующим образом:

1) Пронумеруем переменные подсхемы так, чтобы сначала располагались номера внутренних переменных от 1 до $n-m$, а затем номера граничных переменных от $n-m+1$ до n , и тогда систему (10) можно представить в виде

$$\begin{vmatrix} A_{e,e} & A_{e,d} \\ A_{d,e} & A_{d,d} \end{vmatrix} \begin{vmatrix} X_e \\ X_d \end{vmatrix} = \begin{vmatrix} B_e \\ e_d \end{vmatrix}.$$

2) Исключим из этого уравнения методом Гаусса $n-m$ внутренних переменных. Тогда эта система примет вид

$$\begin{array}{cccc|cccc|c|c} 1 & a_{1,2}^{(1)} & \dots & a_{1,n-m}^{(1)} & a_{1,n-m+1}^{(1)} & \dots & a_{1,n}^{(1)} & & x_{B1} & b_{B1}^{(1)} \\ 0 & 1 & \dots & a_{2,n-m}^{(2)} & a_{2,n-m+1}^{(2)} & \dots & a_{2,n}^{(2)} & & x_{B2} & b_{B2}^{(2)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{n-m,n-m+1}^{(n-m)} & \dots & a_{n-m,n}^{(n-m)} & \times & x_{Bn-m} & b_{Bn-m}^{(n-m)} \\ \hline 0 & 0 & \dots & 0 & a_{n-m+1,n-m+1}^{(n-m)} & \dots & a_{n-m+1,n}^{(n-m)} & & x_{\Gamma n-m+1} & b_{\Gamma n-m+1}^{(n-m)} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \dots & \dots \\ 0 & 0 & \dots & 0 & a_{n,n-m+1}^{(n-m)} & \dots & a_{n,n}^{(n-m)} & & x_{\Gamma n} & b_{\Gamma n}^{(n-m)} \end{array}$$

где верхние символы – номер шага исключения.

Легко видеть, что подсхема, образованная нижней правой частью матрицы A и векторами x_Γ и B_Γ , соответствует выражению (11) и является уравнением подсхемы с исключенными внутренними переменными. Подсхема, образованная верхней трапециевидной матрицей и векторами x_B и B_B , образует уравнение связи граничных и внутренних узлов и соответствует равенству (12). Поскольку верхняя левая подматрица – треугольная с единичными диагональными элементами, то расчет x_B при известных x_Γ выполняется по формулам обратного хода метода Гаусса.

Выражения (11) каждой подсхемы с помощью топологического множества D_Γ объединяются в общую систему уравнений всей схемы относительно только граничных переменных x_Γ :

$$A_\Gamma x_\Gamma = B_\Gamma, \quad (13)$$

размер которой равен мощности множества, полученного объединением множеств x_{Γ} всех подсхем, поэтому оказывается меньше, чем размер полной системы равенств (9) на величину, равную сумме всех исключенных внутренних переменных.

Процесс формирования системы (13) основан на координатном суммировании матриц A_{Γ} и векторов B_{Γ} (где i – номер подсхемы) и состоит в том, что заранее выделяются два массива размерами N_{Γ} и $N_{\Gamma} * N_{\Gamma}$, где N_{Γ} – общее число граничных узлов в схеме, и в эти массивы на позиции, координаты которых определяются оператором D_{Γ} , заносятся составляющие массивов A_{Γ} и B_{Γ} .

Схема может быть настолько большой, что при малом числе подсхем оказывается велик размер систем уравнений (11) и приходится увеличивать количество подсхем. Но тогда увеличивается число граничных переменных x_{Γ} и размер системы (13), который может достигнуть большой величины. В этом случае системе (13) ставится в соответствие некоторая новая схема, состоящая из граничных узлов. В той схеме все внутренние ветви подсхем исходной схемы пересчитаны к граничным узлам, а внутренние узлы, контуры или сечения исключены. По отношению к новой схеме можно применить все преобразования, описанные выше, и получить систему уравнений

$$A^1_{\Gamma} x^1_{\Gamma} = B^1_{\Gamma}, \quad (14)$$

размер которой определяется количеством граничных переменных, оставшихся после исключения части граничных переменных из системы (13).

Далее процесс может быть рекурсивно повторен до тех пор, пока система (14) не окажется допустимых размеров. Таким образом, приходим к идее многоступенчатых подсхем.

Затем решаем систему (14), составленную относительно граничных переменных всей схемы, и с помощью уравнений связи вида (12) находим все внутренние переменные.

4. Алгоритм разбиения схем

Теперь рассмотрим один из методов компоновки частных моделей системы. Задачу разбиения модели системы на подсистемы можно сформулировать как задачу отыскания оптимального разбиения $G = \{G_1, G_2, \dots, G_N\}$ множества элементов схемы $X = \{x_1, x_2, \dots, x_K\}$ на N частей при заданных ограничениях. Некоторые из возможных правил эффективного разбиения приведены в п.1. В литературе [4] встречаются алгоритмы, ориентированные, как правило, на один из критериев, которые оказываются эффективными для определенного класса задач. В качестве примера рассмотрим параллельный алгоритм разбиения схем [2,3], который позволяет сократить время решения путем предварительного выделения максимальных сильносвязанных подграфов и распараллеливания вычислительного процесса. Идея алгоритма последовательного формирования частей разбиения с минимизацией числа внешних связей заключается в следующем. Сначала распределяются специальные элементы или выбранные в качестве специальных. Затем для каждого элемента x_i , принадлежащего множеству всех элементов схемы X , определяется та часть разбиения, при помещении в которую элемент дает наименьшее приращение внешних связей.

Исходной информацией являются:

– матрица цепей $T = [t_{ij}]_{k * c}$, где t_{ij} – номер цепи, связанной с j -м контактом i -го элемента; K – число элементов схемы; c – максимально возможное количество контактов у элементов схемы;

– матрица $S = [s_{ij}]_{n * p}$, где N – число частей разбиения; p – число электрических цепей в схеме; $s_{ij} = 1$, если в i -й части разбиения G_i имеется элемент, инцидентный j -й цепи, и $s_{ij} = 0$ – в противном случае.

Алгоритм состоит в следующем. Для каждого x_i по матрице T строится вектор $S_{0i} = [s_{ij}]_{n * p}$, где $s_{ij} = 1$, если элемент x_i подключен к j -й цепи, и $s_{ij} = 0$ – в противном случае. Возможность назначения i -го элемента в j -ю часть разбиения определяется значением приращения внешних связей dk_{ij} , которое подсчитывается числом единиц в векторе

$$Q = F * \bar{S}_i * S_{0i}, \quad F = \sum_{i=1}^n S_i. \quad (15)$$

Здесь F – вектор, полученный поразрядной дизъюнкцией всех строк матрицы S , кроме j -й, для которой находится ее инверсия. Аналогично подсчитываются dk_{ij} для всех j . Минимальное значение dk_{ij} свидетельствует о целесообразности назначения элемента x_i в подсхему G_i . Процесс повторяется, пока все элементы не будут распределены. Параллельный алгоритм разбиения схем содержит два цикла: внешний – перебора и распределения элементов x_i и внутренний – анализа возможности помещения элемента x_i в j -ю часть разбиения. Внешний цикл можно распараллелить частично путем одновременного анализа группы непосредственно не связанных между собою элементов, что не внесет погрешности в алгоритм.

Процедура выделения максимального подмножества непосредственно не связанных элементов X_H , содержащихся в общем множестве элементов X , в графе, не содержащем изолированные вершины, заключается в следующем.

1. Инвертируем матрицу смежности $R = [r_{ij}]_{k * k}$, где $r_{ij} = 1$, если элемент x_i не связан с элементом x_j , в противном случае $r_{ij} = 0$.

2. По вектор-строке R_i находим множество элементов $C = \{c_i\}$, $i = \overline{1, k_c}$ (где k_c – количество единиц в векторе-строке R_i), не связанных с x_i .

3. Для каждого элемента c_i , принадлежащего C , находим множество не связанных с ним элементов $D = \{D_i(C_i)\}$, $i = \overline{1, k_c}$.

4. Пересечение D_i и R_i дает искомый результат

$$X_H = \left(\prod_{i=1}^{k_c} D_i \right) R_i.$$

Внутренний цикл можно распараллелить путем использования групповых операций и иной организации подсчета значений dk_{ij} . Суть модифицированного алгоритма состоит в следующем.

1. Выделяется группа элементов, непосредственно не связанных между собой и с элементом x_i : $X_H = \{x_j\}$, $i = \overline{1, k_i}$.

2. Для каждого x_i , принадлежащего X_H , по матрице T строится вектор-столбец матрицы $S_0 = [S_{ij}]_{p * k_i}$.

3. Вычисляется вектор

$$F = \sum_{i=1}^p S_i.$$

4. Формируется матрица $Q = [Q_{ij}]_{n * p}$, где

$$Q_i = F \times \bar{S}_i.$$

5. Вычисление приращения внешних связей dk_{ij} , вносимого распределением элемента x_i в G_i для всех $x_i \in X_n$, выполняется одной операцией матричного умножения $dK = Q * S_0$, $dK = \|dk_{ij}\| * k_i$.

6. В результате групповой операции поиска минимального dk_{ij} в каждом столбце матрицы dk определяется целесообразность назначения элемента x_i в j -ю часть разбиения.

7. Находится множество нераспределенных элементов $X' = X \setminus X_n$, и если $X' \neq \emptyset$, пункты 1-7 алгоритма повторяются.

Проведем анализ алгоритмов. Достоинством параллельного алгоритма является снижение на порядок временной сложности по сравнению с базовым вариантом [2]. Аналитически преимущество параллельного алгоритма во времени решения оценим числом операций с векторами: для базового алгоритма — $(3 + N(N+3))Kv$; для параллельного — $(K/v(4v+2N+2Nv))$,

где v — среднее количество элементов X_n .

Таким образом, коэффициент повышения быстродействия параллельного алгоритма

$$a = \frac{(3 + N(N+3))Kv}{(4v + 2N + 2Nv)K} \quad \text{или} \quad a = \frac{N^2 + 3N + 3}{2N(N + N/v) + 4}.$$

Так как минимальное значение $k=2$, а максимальное равно достаточно большому значению N , то нижняя и верхняя оценки

$$a_n = \frac{N^2 + 3N + 3}{3N + 4}, \quad a_b = \frac{N^2 + 3N + 3}{2N + 4}.$$

Недостатком параллельного метода является увеличение количества используемой памяти. Для базового алгоритма требуется $2k_c + Np + 3p + N$ байт оперативной памяти (не считая вспомогательных переменных); для параллельного —

$$2k_c + 2Np + Nv + pv + p,$$

где v — среднее количество элементов X_n . Коэффициент увеличения используемой памяти

$$b = \frac{2k_c + 2Np + Nv + pv + p}{2k_c + Np + 3p + N} = \frac{2k_c + p(2N + v + N) + Nv}{2k_c + p(N + 3) + N}.$$

Задав минимальным значением $v=2$, а максимальным — $v=K$, получим нижнюю и верхнюю оценки

$$b_n = \frac{2k_c + p(2N + 3) + 2N}{2k_c + p(N + 3) + N}, \quad b_b = \frac{2k_c + p(2N + K + 1) + NK}{2k_c + p(N + 3) + N}.$$

Для примера возьмем значения $p=100$, $K=200$, $N=4$, $c=3$, $v_{\min}=2$, $v_{\max}=50$ и вычислим нижние и верхние оценки быстродействия и объема памяти $a_n=1,9$; $a_b=2,5$; $b_n=1,2$; $b_b=3,8$.

Видно, что с увеличением k объем занимаемой памяти растет быстрее, чем быстродействие, в

$$\frac{b_b - b_n}{a_b - a_n} = 4,3 \text{ раза!}$$

Следовательно, возникает противоречие: с одной стороны, для увеличения быстродействия нужно, чтобы количество элементов, непосредственно не связанных между собой, было как можно большим; с другой — выполнение данного условия приводит к резкому увеличению затрат памяти. Самым простым разрешением этого противоречия является наложение ограничений на максимальное значение k , для

приведенного выше примера оно равно 25, тогда $a_b=2,5$, $b_b=2,5$.

При выборе ограничений следует руководствоваться имеющимся объемом памяти и требуемым значением быстродействия с учетом имеющихся распределенных сетевых ресурсов и временных затрат на обмен данными между параллельными вычислительными объектами. Недостатком параллельного метода с точки зрения программирования является необходимость на каждом шаге заново распределять память под массивы S_0 и dK , так как они меняют свой размер в зависимости от v .

Предлагается процедура, реализующая алгоритм разбиения схем с учетом анализа двух описанных выше алгоритмов, которая имеет следующие особенности: минимальное использование оперативной памяти; требуется $2k_c + Np + N$ байт (не учитывая вспомогательных массивов и переменных); иная организация подсчета dk_{ij} , что позволит упростить алгоритм без внесения погрешности и исключить некоторые операции.

Иной подход подсчета dk_{ij} включает: из i -й строки матрицы T берется список связей i -го элемента; в j -й строке матрицы S для j -й части разбиения выбираются элементы с номерами, соответствующими связям i -го элемента, и суммируются:

$$dk_{ij} = \sum_{m=1}^{c_i} S[j, T[i, m]], \quad (17)$$

где c_i — количество контактов i -го элемента. Аналогично рассчитываются dk_{ij} для всех j . Максимальное значение dk_{ij} свидетельствует о целесообразности назначения i -го элемента в j -ю подсхему. “Новое” dk_{ij} показывает, какое количество контактов i -го элемента подключено к уже распределенным цепям в j -й подсхеме. Такое вычисление dk_{ij} (17) позволяет исключить векторы F , S_i , S_{0i} и уменьшить количество операций с ними.

Таким образом, рассмотрен полный круг вопросов, связанных с возможностью и принципами декомпозиции непрерывных систем, предложен один из возможных алгоритмов разбиения и проведена оценка его эффективности.

Литература: 1. Ильин В.Н. Основы автоматизации схемотехнического проектирования. М.: Энергия, 1979. 392с. 2. Морозов К.К., Одинокоев В.Г., Курейчик В.М. Автоматизированное проектирование конструкций РЭА. М.: Радио и связь, 1983. 280 с. 3. Волченская Т.В., Князьков В.С. Параллельный алгоритм компоновки схем// Программирование. 1991. №1. С. 94-95. 4. Мороз Б.И. Декомпозиция сети ЭВМ коллективного пользования сложной структуры с целью применения моделей управления распределением ресурсов/ АСУ и приборы автоматки. 1982. Вып.61. С. 58-61.

Поступила в редколлегию 12.02.98

Горбачев Валерий Александрович, канд. техн. наук, профессор кафедры ЭВМ ХТУРЭ. Научные интересы: теория моделирования. Адрес: 310726, Украина, Харьков, пр. Ленина, 14, тел. 40-93-54.

Волк Максим Александрович, аспирант, младший научный сотрудник кафедры ЭВМ ХТУРЭ. Научные интересы: теория проектирования электронных систем, теория моделирования. Адрес: 310726, Украина, Харьков, пр. Ленина, 14, тел. 40-93-54.

Бабаев Андрей Павлович, аспирант, младший научный сотрудник кафедры ЭВМ ХТУРЭ. Научные интересы: теория систем, теория моделирования. Адрес: 310726, Украина, Харьков, пр. Ленина, 14, тел. 40-93-54.