

**Применение операций над гиперграфами
для компоновки схем ЭВМ**

07, июль 2011

авторы: Овчинников В. А., Пламядялов А. А.

УДК 004

УДК 004.3 + 519.6

МГТУ им. Н.Э. Баумана
vaovchinnikov@gmail.com

Задача компоновки состоит в определении состава конструктивных модулей ЭВМ. Рассмотрим решение задачи компоновки декомпозицией структуры объекта, например функциональной схемы, на части таким образом, чтобы критерий оптимальности достигал экстремального значения. Корректной моделью схемы является гиперграф, представленный в форме $H(X, U, GX, GU, F_1X)$ или кусок гиперграфа $H^k(X^k, U^k, GX^k, GU^k, F_1X^k)$, а задача декомпозиции состоит в разрезании $H(X, U)$ или $H^k(X^k, U^k)$ на совокупность $B(H_l^k)$ кусков $H_l^k(X_l, U_l)$, $l = 1, L$, где L – требуемое количество частей структуры. Будем считать, что моделью схемы является кусок гиперграфа H^k , который задан аналитически в форме $H^k(X^k, U^k, GX^k, GU^k, F_1X^k)$.

Описанные в литературе алгоритмы декомпозиции решают задачу разбиения множества вершин гиперграфа X или его куска X^k на совокупность непересекающихся подмножеств $X_l - B(X_l) = \{X_l\}$, $l = 1, L$ таких, что:

$$X_l \neq \emptyset, X_l \cap X_p = \emptyset, \bigcup_{l \in L} X_l = X.$$

В процессе автоматизированного проектирования после определения множества X_l каждого куска необходимо получить модель соответствующей подсхемы и удалить её из модели схемы, т. е.:

– определить множество рёбер куска U_l , разделить его на подмножества внутренних U_{int} и внешних U_{ext} ;

– определить множества образов вершин ΓX_l и рёбер ΓU_l полученного куска относительно предикатов инцидентности и множество образов вершин $F_1 X_l$ относительно предиката смежности;

– удалить множества X_l , U_l , ΓX_l , ΓU_l и $F_1 X_l$ из множеств X^k , U^k , ΓX^k , ΓU^k и $F_1 X^k$ соответственно.

Использование операций над гиперграфами позволяет получить результат в виде L кусков гиперграфов, являющихся моделью схемы, заключенной в соответствующей вершине дерева функционального вхождения, т. е. реализовать описанную часть процесса компоновки, сделав процесс решения задачи более наглядным

Основными процедурами, используемыми проектировщиком при решении задачи компоновки, являются: выделение части схемы и удаление её из разрезаемой схемы. Данным проектным процедурам соответствуют операции формирования куска гиперграфа, объединение кусков и дополнения куска до гиперграфа или его куска [1, 2].

Приведем последовательный алгоритм разрезания гиперграфа схемы в двух вариантах: в операциях над множествами и с использованием операций над гиперграфами. Оценим вычислительную сложность алгоритма и проведем сравнение с вычислительной сложностью того же алгоритма, описанного только с применением операций над множествами. При описании исключим оптимизирующие преобразования, которые различны для одного и другого случая и вносят в вычислительную сложность коррективы, подлежащие дополнительному изучению. Для краткости в алгоритме будем использовать упрощённую запись операций над гиперграфом. Например, формирование одновершинного куска с заранее заданной вершиной x_q выполняется операцией $H_l^k(X_l, U_l, \Gamma_1 X_l, \Gamma_2 U_l, F_1 X_l) := H^k(X, U, \Gamma_1 X, \Gamma_2 U, F_1 X) \odot x_q$, которую в алгоритме будем записывать сокращённо как $H_1^k := H^k \odot x_q$.

Ниже в левом столбце табл. 1 приведено описание последовательного алгоритма решения этой задачи в операциях над множествами, в правой – с использованием указанных выше операций над гиперграфами.

Таблица 1

| Описание алгоритма в операциях над множествами | Описание алгоритма с использованием операций над графами |
|--|---|
| <p>1. Включаем в формируемое множество X_l заранее заданную вершину:</p> $X_l := \{x_q\};$ | <p>Формируем одновершинный кусок $H_l^k := H^k \odot x_q$ и удаляем его из модели схемы $H^k := H^k \setminus H_l^k$;</p> |
| <p>2. Определяем количество рёбер, соединяющих x_q с $X \setminus x_q$:</p> $S_0 := \Gamma_1 x_q ;$ | <p>Без изменений</p> |
| <p>3. Находим множество X_k вершин – кандидатов на включение в X_l как множество вершин, смежных вершине x_q:</p> $X_k := F_1 x_q;$ | <p>Без изменений</p> |
| <p>4. Организуем цикл определения для каждой вершины $x_i \in X_k$ множества U_i^n и количества Δs_i^+ рёбер, приходящих в разрез между кусками, множество вершин которых X_l и $X \setminus X_l$, и Δs_i^- – количества рёбер уходящих из этого разреза:</p> $(\forall x_i \in X_k):$ | <p>Без изменений</p> |
| <p>4.1. Находим множество рёбер, инцидентных вершине x_i, и осуществляем инициализацию параметров $U_i^n, \Delta s_i^-, \Delta s_i^+$:</p> $U_i := \Gamma_1 x_i; U_i^n := \emptyset; \Delta s_i^- := 0; \Delta s_i^+ := 0;$ | <p>Без изменений</p> |
| <p>4.2. Организуем цикл анализа рёбер множества U_i:</p> $(\forall u_j \in U_i):$ | <p>Без изменений</p> |
| <p>4.2.1. Определяем вершины, инцидентные ребру u_j:</p> $X_j := \Gamma_2 u_j;$ | <p>Без изменений</p> |
| <p>4.2.2. Проверяем условие $(\forall x_k \in X_j) (x_k \in X_l \cdot x_i) = \text{«истина»}$. Если условие удовлетворяется, то переход к п. 4.2.3, иначе – к п. 4.2.4;</p> | <p>Без изменений</p> |
| <p>4.2.3. Подсчитываем показатель $\Delta s_i^- := \Delta s_i^- + 1$ и переходим к п. 4.2</p> | <p>Без изменений</p> |
| <p>4.2.4. Проверяем условие $(\forall x_k \in X_j) (x_k \notin X_l) = \text{«истина»}$. Если условие удовлетворяется, то переход к п. 4.2.5, иначе – к п. 4.2;</p> | <p>Без изменений</p> |

| | |
|---|--|
| 4.2.5. Запоминаем ребро, приходящее в разрез, и подсчитываем показатель Δs_i^+ : $U_i^n := U_i^n \bullet u_j$; $\Delta s_i^+ := \Delta s_i^+ + 1$; | <i>Без изменений</i> |
| 4.3. Подсчитываем показатель $\Delta s_i := \Delta s_i^+ - \Delta s_i$; | <i>Без изменений</i> |
| 5. Находим вершину $x_t \in X_K$, для которой Δs минимально: $\Delta s_t := \min \{ \Delta s_i \in \Delta S \} ; x_t \leftrightarrow s_t$; | <i>Без изменений</i> |
| 6. Подсчитываем количество рёбер в разрезе: $S_0 := S_0 + \Delta s_t$; | <i>Без изменений</i> |
| 7. Проверяем ограничение на количество внешних выводов l -й части схемы $S_0 \leq [s_l]_{\text{доп}}$. Если условие выполняется, то переход к п. 8, иначе – к п. 12; | <i>Без изменений</i> |
| 8. Включаем вершину в множество $X_l := X_l \bullet x_t$; | <i>Формируем одновершинный кусок $H_l^k := H^k \odot x_t$, добавляем его к модели l-й части схемы $H_l^k := H_l^k \cup H_l^k$ и удаляем из модели оставшейся части схемы $H^k := H^k \setminus H_l^k$;</i> |
| 9. Проверяем условие $ X_l < n_l$, где n_l – требуемое количество элементов l -й части схемы. Если условие выполняется, переход к п. 10, иначе – к п. 13; | <i>Без изменений</i> |
| 10. Определяем множество вершин, инцидентных рёбрам множества U_i^n : $X_t^n := \cup \Gamma_2 u_j ;$ $u_j \in U_i^n$ | <i>Без изменений</i> |
| 11. Корректируем множество вершин $X_K := \{X_K \cup X_t^n\} \setminus x_t$ и возвращаемся к п. 4; | <i>Без изменений</i> |
| 12. Дальнейшее формирование невозможно из-за нарушения ограничений, переход к п. 14; | <i>Без изменений</i> |
| 13. Вывод результатов – подмножество вершин X_l . | <i>кусок гиперграфа H_l^k.</i> |
| 14. Конец алгоритма. | |

Как видим, для получения куска гиперграфа непосредственно алгоритмом использованы следующие операции над гиперграфом: формирование одновершинного куска, удаление его из модели оставшейся части схемы

(в п.п. 1 и 8) и объединение одновершинного куска с куском – моделью формируемой части схемы (п. 8).

Получим оценку вычислительной сложности процесса выделения одной подсхемы для обеих версий алгоритма. При использовании алгоритма в операциях над множествами вычислительная сложность процесса складывается из сложности алгоритма, операции формирования куска гиперграфа по множеству $X_l^k - H_l^k(X_l, U_l) := H^k(X^k, U^k) \odot X_l$ и операции дополнения сформированного куска гиперграфа $H_l^k(X_l, U_l)$ до куска гиперграфа $H^k(X^k, U^k)$ (модели оставшейся части схемы) $H^k(X^k, U^k) := H^k(X^k, U^k) \setminus H_l^k(X_l, U_l)$.

Асимптотическая оценка вычислительной сложности первой версии алгоритма получена в [3] и равна $O(n^{5/2})$. Вычислительная сложность операции формирования куска гиперграфа равна $O(m^2)$ или $O(n \cdot m)$, так как $|X_l|$ ограничена $n_l = n / const$, $|U_l| = m / const$, $|\Gamma x_i| = const$ и $|\Gamma u_j| = const$ (см. статья 2 в эл. Ж.). Учитывая, что для схем ЭВМ, представленных в функциональных элементах невысокой степени интеграции, $n \approx m$ с точностью до константы, будем использовать оценку $O(n^2)$.

Для рассматриваемой задачи схемно-топологического проектирования средств ЭВТ асимптотическая оценка вычислительной сложности операции дополнения «в худшем» – существенно завышена. Представляется целесообразным получить реалистичную оценку. Анализ формального описания этой операции [2] показывает, что определяющий вклад в вычислительную сложность вносит определение образов ребер куска – результата операции. С учётом принятых обозначений соответствующее правило примет вид:

$$\Gamma U_l = \{\Gamma u_j : u_j \in U_{\text{int}} \vee \Gamma u_j \setminus X_l : u_j \in U_{\text{ext}} / u_j \in U_l, \Gamma u_j \in \Gamma U^k\}.$$

Так как $|U_{\text{int}}| < m$ и согласно закону Рента [4] $|U_{\text{ext}}|$ ограничена $n^{1/2}$, реалистичной оценкой вычислительной сложности операции дополнения будет $O(n^{3/2})$. Приводя все оценки к переменной n , получаем:

$$O_{\text{рез}} = O(n^{5/2}) + O(n^2) + O(n^{3/2}) = O(n^{5/2}).$$

Для реализации алгоритма с использованием операций над гиперграфами сопоставляемой с $O_{рез}$ величиной является вычислительная сложность самого алгоритма. Пунктами, имеющими различную вычислительную сложность, являются пункты 1 и 8.

Операция формирования одновершинного куска гиперграфа $H_1^K(X_1, U_1)$ имеет вычислительную сложность $O(1)$, дополнения его до куска $H^K - O(n^2 \cdot m)$ в худшем. Получим реалистичную оценку вычислительной сложности, учитывая, что в операции участвует одновершинный кусок гиперграфа. Для этого правило определения образов ребёр куска гиперграфа – результата этой операции запишем в виде:

$$\Gamma\{U^K \setminus U_{1int}\} = \{\Gamma u_j : u_j \notin \{U_{ext}^K \cdot U_1\} \vee \Gamma u_j \setminus X_1 : u_j \in U_{ext}^K / u_j \in \{U^K \setminus U_{1int}\}, \Gamma u_j \in \Gamma U^K\}.$$

Так как $|U_{ext}^K|$ ограничена $n^{1/2}$, $|U_1| < \rho(u_j) \cdot \rho(x_i) = const$ и $|X_1^K| = 1$, асимптотическая оценка вычислительной сложности данной операции будет $O(n^{3/2})$.

Вычислительная сложность операции объединения куска гиперграфа с одновершинным куском равна $O(n)$. Все эти три операции в п. 8 выполняются $(n_l - 1)$ раз. Очевидно, что наибольший вклад в вычислительную сложность внесёт операция дополнения куска до куска в п. 8, который оценивается величиной $O(n^{3/2}) \cdot O(n) = O(n^{5/2})$.

Как видно из выполненного анализа асимптотические оценки вычислительной сложности проектной процедуры выделения одной подсхемы для обоих вариантов реализации процесса компоновки одинаковы.

Напомним, что результатом работы алгоритма в варианте без операций над гиперграфами является подмножество вершин куска гиперграфа X_l^K , а с использованием операций формирования, объединения и дополнения – аналитическое представление куска в форме $H_l^K(X_l, U_l, \Gamma_1 X_l, \Gamma_2 U_l, F_1 X_l)$.

Использование операций над гиперграфами рассмотрим также на примере формализации многоуровневого (блочного-иерархического) подхода, реализованного в программном комплексе компоновки hMeTiS [5]. Данный программный комплекс предназначен для разрезания гиперграфов большого размера и может быть использован при проектировании СБИС, баз данных, систем управления транспортом и т.п. Задача компоновки ставится так же,

как и выше. В простейшей версии основная процедура реализует проектные операции укрупнения модели, её начального разрезания, улучшения этого разрезания и разукрупнения модели, организованные по следующей схеме.

Сначала посредством многократного применения алгоритма свертки вершин гиперграфа сокращают их количество до 100...200 штук. Таким образом создаются предпосылки для многократного преобразования модели объекта с разной степенью его детализации.

Выполняется начальное разбиение множества вершин гиперграфа или его куска – модели схемы на два примерно равных по мощности подмножества. Это разбиение осуществляется случайным или последовательным алгоритмом.

Далее на каждом уровне детализации, начиная с последнего, выполняется разукрупнение вершин гиперграфа, разрезание улучшается итерационным алгоритмом. Эти две процедуры повторяются до тех пор, пока не будет получено разрезание исходного гиперграфа $H(X, U)$ на две части, т. е. на два куска $H_1^k(X_1, U_1)$ и $H_2^k(X_2, U_2)$.

Описанная основная процедура применяется к полученным кускам гиперграфа до получения совокупности $B(H_i^k)$ кусков $H_i^k(X_i, U_i)$ с требуемым количеством вершин.

Для снижения размера гиперграфа могут использоваться алгоритмы уравновешенной двоичной или n -арной (неуравновешенной и уравновешенной) свертки. При n -арной свертке объединяются вершины не смежных рёбер. Результатом работы алгоритма свертки на r -м уровне детализации является аналитическое представление гиперграфа $H(Y^r, V^r)$, где $y_j^r = Y_j^{r-1}_{\text{св}} \subseteq Y^{r-1}$ и множество кусков $\{H_j^k(Y_j^{r-1}_{\text{св}}, V_j^{r-1}_{\text{св}})\}, j = 1, 2, \dots, |Y^r|$ (после первого применения алгоритма свертки, т.е. на первом уровне детализации, получается гиперграф $H(Y^1, V^1)$, где $y_j^1 = X_{j\text{св}} = \{x_i, x_k, \dots, x_l\} \subseteq X$).

Рассмотрим вариант, когда результатом работы алгоритмов начального разрезания и итерационного улучшения являются два подмножества вершин, соответствующего уровня. Для работы всех указанных алгоритмов гиперграф целесообразно задавать множествами вершин, рёбер и их образами относительно предикатов инцидентности и смежности. Как было условлено в разде-

ле 3 для краткости исходный гиперграф и гиперграфы каждого уровня детализации будем обозначать в виде $H(X, U)$ и $H(Y^r, V^r)$.

Для формального описания процесса интерес представляют этапы, следующие за свёрткой. Для r -го уровня детализации это описание с использованием операций над гиперграфами представлено в табл. 2. Здесь после обозначения алгоритма в круглых скобках указан вход, а после символа « \rightarrow » – результат работы алгоритма.

Таблица 2

| N | Выполняемый алгоритм или операция | Комментарии |
|----|--|--|
| 1. | $A_{\text{д}}(H(Y^r, V^r)) \rightarrow \{Y_1^r, Y_2^r\}$ | <i>Разбиение множества Y^r на два подмножества случайным или последовательным алгоритмом</i> |
| 2. | $H_1^K(Y_1^r, V_1^r) = H(Y^r, V^r) \boxtimes Y_1^r,$ $H_2^K(Y_2^r, V_2^r) = H(Y^r, V^r) \boxtimes Y_2^r$ | <i>Получение полного аналитического представления кусков H_1^K и H_2^K на r-ом уровне детализации операцией формирования</i> |
| 3. | $(\forall y_j \in Y_1^r):$ $H_1^K(Y_1^{r-1}, V_1^{r-1}) = H_1^K(Y_1^r, V_1^r / y_j \Rightarrow Y_{\text{св}}^{r-1}),$ $(\forall y_j \in Y_2^r):$ $H_2^K(Y_2^{r-1}, V_2^{r-1}) = H_2^K(Y_2^r, V_2^r / y_j \Rightarrow Y_{\text{св}}^{r-1}),$ | <i>Получение аналитического представления кусков H_1^K и H_2^K на $r-1$-ом уровне детализации применением операции дефакторизации к вершинам множеств Y_1^r и Y_2^r</i> |
| 4. | $H(Y^{r-1}, V^{r-1}) = H_1^K(Y_1^{r-1}, V_1^{r-1}) \cup H_2^K(Y_2^{r-1}, V_2^{r-1})$ | <i>Получение аналитического представления гиперграфа на $r-1$-ом уровне детализации операцией объединения кусков</i> |
| 5. | $A_{\text{ит}}(H_1^K(Y_1^{r-1}, V_1^{r-1}), H_2^K(Y_2^{r-1}, V_2^{r-1})) \rightarrow \{Y_{1y}^{r-1}, Y_{2y}^{r-1}\}$ | <i>Улучшение разрезания итерационным алгоритмом</i> |
| 6. | $H_1^K(Y_1^{r-1}, V_1^{r-1}) = H(Y^{r-1}, V^{r-1}) \boxtimes Y_{1y}^{r-1},$ $H_2^K(Y_2^{r-1}, V_2^{r-1}) = H(Y^{r-1}, V^{r-1}) \boxtimes Y_{2y}^{r-1}$ | <i>Получение аналитического представления кусков H_1^K и H_2^K на $r-1$-ом уровне детализации операцией формирования после перестановки вершин</i> |

При известных оценках вычислительной сложности используемых алгоритмов и операций над гиперграфами полученное формальное описание позволяет легко получить результирующую оценку и выявить преобразования, вносящие наибольший вклад в неё.

Литература

1. Овчинников В.А. **Операции над ультра- и гиперграфами для реализации процедур анализа и синтеза структур сложных систем:** Наука и образование. Инженерное образование: Эл. Науч. Издание. – 2009. – № 11.
2. Овчинников В.А. **Операции над ультра- и гиперграфами для реализации процедур анализа и синтеза структур сложных систем:** Наука и образование. Инженерное образование: Эл. Науч. Издание. – 2009. – № 12.
3. Овчинников В.А. **Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем:** МГТУ им. Баумана, 2001, – 288 с.
4. Савельев А. Я., Овчинников В.А. **Конструирование ЭВМ и систем:** Учеб. для вузов по спец. «Выч. маш., компл., сист. и сети.» – 2-е изд., перераб. и доп. – М.: Высш. шк., – 312с.
5. G. Karypis, R. Aggarwal, V Kumar, S. Shekhar, «**Multilevel Hypergraph Partitioning: Applications in VLSI Domain**», *IEEE Trans. VLSI Syst.*, vol. 7, NO 1, March 1999, pp. 69-79.