

Оценка эффективности применения операций над упорядоченными множествами**77-30569/239406**

10, октябрь 2011

Овчинников В. А., Иванова Г. С.

УДК 004.3 + 519.6

МГТУ им. Н.Э. Баумана

gsivanova@gmail.com

Эффективность применения операций над упорядоченными множествами рассмотрим на примере решения задачи декомпозиции структуры сложной системы на подсистемы, например схемы электрической функциональной на подсхемы. Известна формальная постановка этой задачи как задачи разбиения множества вершин гиперграфа на совокупность непересекающихся подмножеств [1]. Там же рассмотрен и один из вариантов её решения посредством алгоритма последовательного разрезания гиперграфа. Результатом работы этого алгоритма является множество вершин куска гиперграфа $X_l \leftrightarrow \mathcal{E}_l$, где $\mathcal{E}_l$ – множество компонентов сформированной подсистемы.

Множество вершин $X_l \leftrightarrow \mathcal{E}_l$ может быть также получено, например алгоритмом выделения минимальных массивов гиперграфа [2]. С практической точки зрения предпочтительней выделять квазимиимальные массивы с ограничением $\min \leq |X_l| \leq \max$.

После окончания процесса декомпозиции необходимо проверить условия:

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

Оценку эффективности использования операций над упорядоченными множествами выполним на примере формирования множества X_l последовательным алгоритмом. Это объясняется тем, что алгоритм выделения

минимальных массивов гиперграфа сложен, а вычислительная сложность его реализации зависит от распределения связей между компонентами объекта и может быть в пределе экспоненциальной.

После определения X_l основными проектными процедурами, реализующими процесс декомпозиции, являются: выделение части схемы, удаление её из разрезаемой схемы и повторный запуск алгоритма. Включим в процесс также контрольную операцию:

$$\forall i = 1(|X_l| - 1)(x_i \neq x_{i+1}),$$

проверяющую возможность такого искажения информации, при которой один и тот же компонент системы будет указан неоднократно.

Таким образом, процесс декомпозиции структуры системы составят следующие процедуры:

1. Определение $X_l^k = X_l$ последовательным алгоритмом разрезания гиперграфа.

2. Проверка условия

$$\forall i = 1(|X_l| - 1)(x_i \neq x_{i+1}).$$

3. Формирование куска $H_l^k(X_l^k, U_l^k)$ гиперграфа.

4. Удаление куска $H_l^k(X_l^k, U_l^k)$ из модели системы (гиперграфа после первого применения алгоритма разрезания и куска гиперграфа в дальнейшем).

5. Проверка условия окончания процесса декомпозиции $|X_{\text{ост}}^k| > n_l$, где $X_{\text{ост}}^k$ – множество оставшихся вершин гиперграфа (ещё не скомпонованных элементов системы), n_l – допустимое количество компонентов подсистемы. При выполнении условия – переход к п. 1, в противном случае – к п. 6.

6. Проверка условий

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

Получим оценку сверху вычислительной сложности реализации описанного процесса в функции от мощности множеств X , (X_l^k) , U , (U_l^k) , G_x и G_u , учитывая только операции сравнения.

Приведём описание и схему первого варианта последовательного алгоритма разрезания гиперграфа [3]. Гиперграф задан аналитически в форме $H(X, U, \Gamma X, \Gamma U, F_1 X)$. На текущем шаге алгоритма в качестве кандидатов на включение в формируемое подмножество X_l выступают вершины, связанные с вершинами, уже вошедшими в X_l . Обозначим это множество через X_k .

Количество рёбер, попадающих в разрез, подсчитывается по рекуррентной формуле:

$$s_i = s_0 + \Delta s_i^+ - \Delta s_i^-,$$

где s_0 – количество ребер, находящихся в разрезе между кусками гиперграфа после выполнения предыдущего шага алгоритма, Δs_i^+ и Δs_i^- – количество рёбер приходящих в разрез и уходящих из него при включении в X_l вершины x_i . Из рёбер $u_j \in \Gamma x_i$, т. е. инцидентных вершине x_i , приходят в разрез те, для которых выполняется условие:

$$X_j \cap X_l = \emptyset, \text{ где } X_j = \Gamma u_j, \quad (1)$$

и уходят из него такие, для которых справедливо:

$$X_j \setminus x_i \subseteq X_l. \quad (2)$$

Множество вершин кандидатов X_k корректируется, исключая из него вершину, вошедшую в X_l , и добавляя новые, которые становятся кандидатами за счет включения в разрез новых ребер – это множество X_t^n в п.10 алгоритма. Структуры данных для представления всех множеств и массива $\Delta S = \langle \Delta s_i \rangle$ – динамические вектора.

1. Включаем в формируемое множество X_l некоторую вершину x_q :

$$X_l = \{x_q\}.$$

2. Определяем количество ребер s_0 , соединяющих x_q с $X \setminus x_q$:

$$U' = \Gamma x_q, s_0 = |U'|.$$

3. Находим множество вершин X_k – кандидатов на включение в X_l :

$$X_k = F_1 x_q.$$

4. Для каждой вершины $x_i \in X_k$ определяем множество инцидентных ей ребер U_i , подмножество $U_i^{\Pi} \subset U_i$ ребер, приходящих в разрез, и подсчитываем показатель Δs_i . Для чего выполняем:

$$4.1. U_i = \Gamma x_i, \Delta s_i^- = 0, \Delta s_i^+ = 0.$$

4.2. Для $\forall u_j \in U_i$ выполняем п.п. 4.2.1...4.2.5:

4.2.1. Находим множество вершин X_j , входящих в ребро u_j : $X_j = \Gamma u_j$.

4.2.2. Проверяем условие

$$X_j \setminus x_i \subseteq X_i = \text{"истина"}.$$

Если условие не выполняется, то ребро u_j находится в разрезе – переходим к п.п. 4.2.4. Выполнение условия означает, что ребро уходит из разреза – переходим к п. п. 4.2.3.

4.2.3. Подсчитываем показатель Δs_i^- : $\Delta s_i^- = \Delta s_i^- + 1$

и переходим к п. 4.2.

4.2.4. Проверяем условие $X_j \cap X_i = \emptyset = \text{"истина"}$.

Если условие выполняется (это означает, что ребро u_j приходит в разрез при включении x_i в X_i), то переходим к п.п. 4.2.5. При невыполнении условия переходим к анализу следующего ребра, т.е. к п. 4.2.

4.2.5. Включаем ребро u_j в множество U_i^{Π} : $U_i^{\Pi} = U_i^{\Pi} \cup u_j$, подсчитываем показатель Δs_i : $\Delta s_i^+ = \Delta s_i^+ + 1$ и возвращаемся в п. 4.2.

4.3. Определяем значение Δs_i : $\Delta s_i = \Delta s_i^+$.

5. Находим вершину $x_t \in X_k$, для которой Δs минимально:

$$\Delta s_t = \min \{ \Delta s_i \in \Delta S \}, x_t \leftrightarrow \Delta s_t.$$

6. Подсчитываем количество ребер S_0 , соединяющих множества X_l и $X \setminus X_l$ после включения x_t в X_l :

$$S_0 = S_0 + \Delta s_t.$$

7. Проверяем ограничение на количество внешних выводов l -й части схемы: $S_0 \leq [S_l]_{\text{доп}}$.

Если условие выполняется, то переходим к п.8, иначе – к п. 12.

8. Включаем вершину x_t в множество X_l , и исключаем ее из X_k :

$$X_l = X_l \cup x_t, X_k = X_k \setminus x_t.$$

9. Проверяем условие: $|X_l| < n_l$,

где n_l – требуемое количество элементов в l -ой части схемы. Если условие выполняется, то переходим к п.10, иначе – к п.13.

10. Определяем множество вершин, входящих в множество ребер U_t^n :

$$X_t^n = \bigcup_{u_j \in U_t^n} \Gamma u_j \setminus x_t.$$

11. Корректируем множество вершин кандидатов X_k :

$$X_k = \{X_k \cup X_t^n\}$$

и возвращаемся к п.4.

12. Дальнейшее формирование X_l невозможно из-за нарушения ограничения на число внешних выводов. Переход на окончание работы алгоритма к п. 14.

13. Вывод результатов.

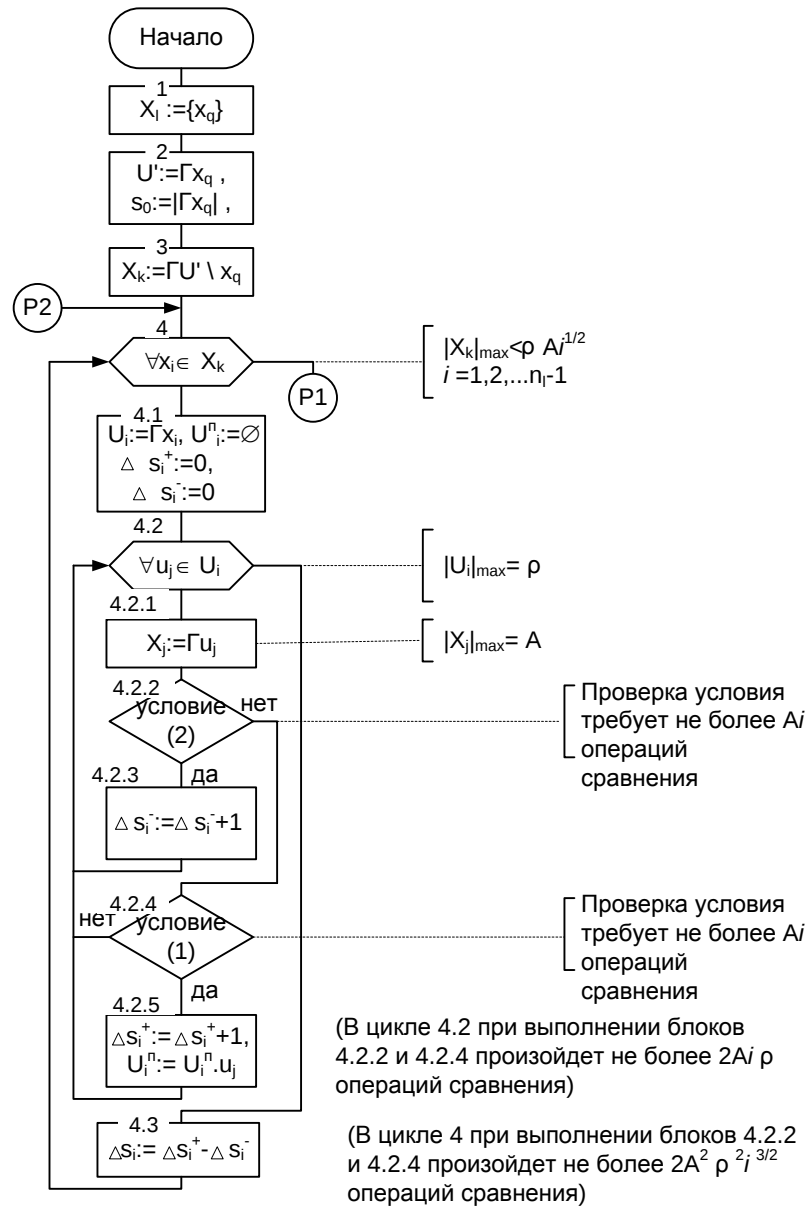
14. Конец работы алгоритма.

Схема изложенного алгоритма показана на рис. 1, нумерация пунктов описания и блоков схемы совпадает.

На схеме в комментариях показаны размерности массивов (множеств) и количества операций сравнения, выполняемых в ходе работы алгоритма. Все оценки, кроме размера множеств X_k (и массива ΔS) и количества операций сравнения, необходимых для нахождения массива X_t^n в п. 10, очевидны. Максимальное количество вершин кандидатов $|X_k|_{\max}$ оценим, используя закон Рента [2]. На основании этого закона количество внешних выводов схемы от числа элементов в ней оценивается по формуле $N_{\text{в.}} = \alpha \cdot N^p$, где α – среднее количество входных и выходных выводов элементов схемы, $p = 0, 5 \dots 0,75$ – показатель Рента. В наших обозначениях α – это $|\Gamma x| = \rho$. Примем α равным $\rho_{\max}$ (в дальнейшем будем $\rho_{\max}$ обозначать просто ρ) и $p = 0,5$. Тогда количество ребер в разрезе $|U'|_{\max} = \rho \cdot |X_l|^{\frac{1}{2}}$, где $|X_l| = i = 1, 2 \dots n_l$.

Так как в каждое ребро входит не более A вершин, то

$$|X_k|_{\max} \leq \rho \cdot A \cdot i^{1/2}.$$



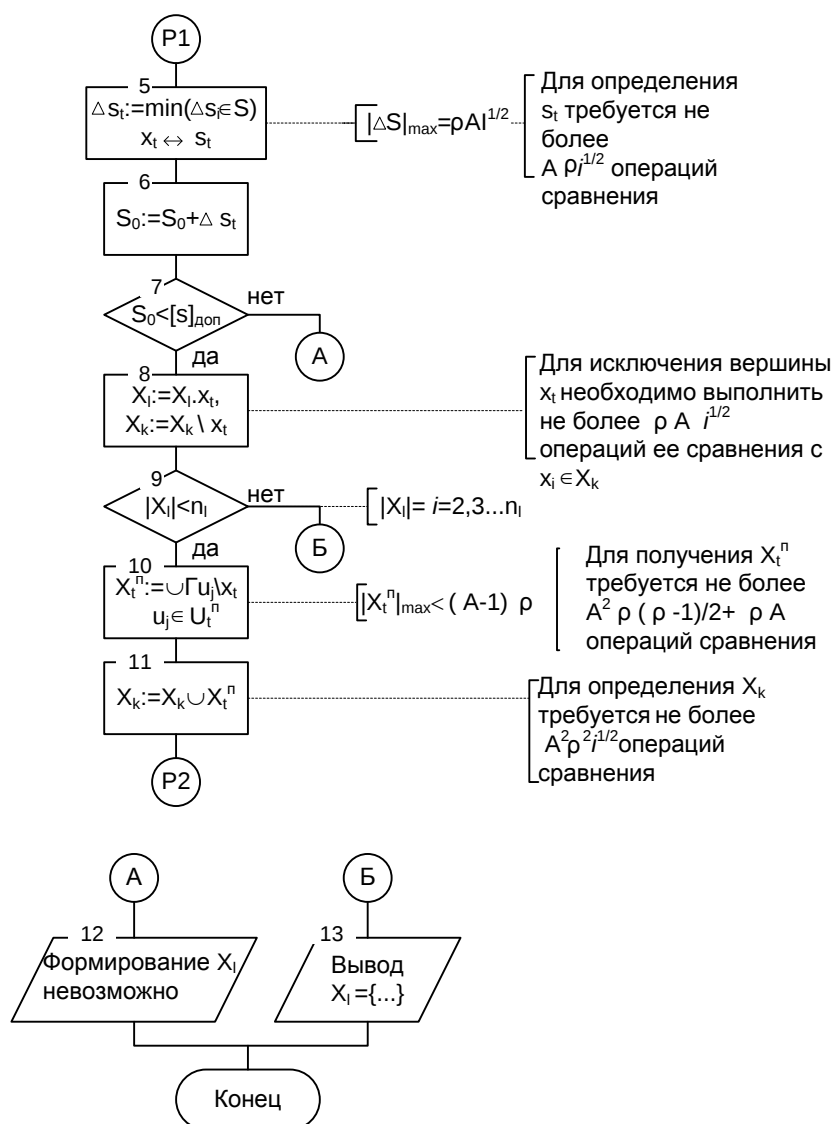


Рис. 1. Схема последовательного алгоритма разрезания гиперграфа

Для сравнительной оценки вычислительной сложности реализации процесса декомпозиции достаточно рассмотреть однократное выполнение процедур 1...4 и процедуру 6. Отметим, что при использовании операций над упорядоченными множествами сортировке подлежат только множество X_i в порядке возрастания индексов элементов, массив ΔS по возрастанию значений его элементов и элементы x_i множества X_k в соответствии с положением Δs_i в массиве ΔS .

1. Определение $X_i^K = X_i$ последовательным алгоритмом разрезания гиперграфа. Основной вклад в вычислительную сложность алгоритма вносит его циклическая часть – п. п. 4...11.

Выполнение цикла 4.

Неупорядоченные множества. Проверка условий блоков 4.2.2 и 4.2.4 требует не более $2A \cdot i$ операций сравнения элементов массивов X_j и X_l , следовательно, вычислительная сложность цикла 4 не более

$$2A \cdot i \cdot \rho \cdot \rho \cdot A \cdot i^{1/2} = 2A^2 \cdot \rho^2 \cdot i^{3/2}.$$

Упорядоченные множества. Проверка условий блоков 4.2.2 и 4.2.4 на основании (2) и (1) требует не более $2i$ и $2(i + A - 1)$ операций сравнения соответственно. Отсюда вычислительная сложность реализации п. 4 будет не более

$$2(2i + A) \cdot \rho \cdot \rho \cdot A \cdot i^{1/2} = 4A \cdot \rho^2 \cdot i^{3/2} + 2A \cdot \rho^2 \cdot i^{1/2}.$$

Как видно из схемы алгоритма количество операций сравнения, необходимых для выполнения п. п. 5, 8 и 11 имеет на единицу меньший порядок сложности.

Оценим вклад, вносимый процедурой сортировки при упорядочивании множеств X_k и X_l . После выполнения п.3 $|X_k| < \rho \cdot A$. При каждом i – м выполнении тела цикла 4 удаляем одну вершину и 3 $|X_t^n| < \rho \cdot A$ вершин добавляем. Ориентируясь на сортировку с помощью двоичной кучи, получим для X_k

$$N_{\Sigma} = \rho \cdot A \cdot \log_2(\rho \cdot A) + \rho \cdot A \sum_{i=1}^{n_l-1} \log_2(\rho \cdot A \cdot i^{1/2}),$$

где N_{Σ} – суммарное количество операций сравнения, выполняемых при формировании множества X_l . Найдя через интеграл оценку

$$\sum_{i=1}^n \log_2 i \cong n \log_2 n - n,$$

имеем

$$N_{\Sigma} < \rho \cdot A \cdot n_l [\log_2(\rho \cdot A) + (\log_2 n_l - 1)/2].$$

Упорядочивание множества X_k требует $n_l \log_2 n_l$ операций сравнения. Таким образом, вклад процедур упорядочивания также имеет меньший порядок, чем цикла 4. Оценивая эффективность использования операций над

упорядоченными множествами при получении X_l последовательным алгоритмом как отношение членов старшего порядка вычислительной сложности цикла 4, получим:

$$K_{\text{алг}} = 2A^2 \cdot \rho^2 \cdot i^{3/2} / (4A \cdot \rho^2 \cdot i^{3/2}) = A / 2.$$

2. Проверка условия $\forall i = 1 (|X_l^K| - 1) (x_i \neq x_{i+1})$. Очевидно, что проверяя это условие, мы должны предполагать, что X_l^K может быть неупорядоченным кортежем. Например, в процессе упорядочивания множества $X_l^K = \{x_3, x_1, x_6, x_4, x_5, x_9\}$ из-за искажения имени элемента x_9 на x_4 может быть получен кортеж $\langle x_1, x_3, x_4, x_5, x_6, x_4 \rangle$. Следовательно, в алгоритме, устанавливающем истинность или ложность этого выражения, никакие свойства упорядоченных множеств использовать нельзя.

3. Формирование куска $H_l^K(X_l^K, U_l^K)$ гиперграфа. При оценке вычислительной сложности выполнения этого этапа процесса декомпозиции будем считать, что кусок гиперграфа должен быть получен в форме $H_l^K(X_l^K, U_l^K, \Gamma X_l^K, \Gamma U_l^K)$. Подсчёт количества операций сравнения будем выполнять, используя выражения формального описания выполнения операции получения куска H_l^K гиперграфа [4].

3.1. Определение множества

$$U_l^K = \bigcup_{x_i \in X_l^K} \Gamma x_i, \Gamma x_i \in \Gamma X,$$

Неупорядоченные множества. Количество операций сравнения, необходимых для определения U_l^K будет определяться как:

$$N_{3.1} = \rho^2 \cdot \sum_{i=1}^{|X_l^K|} [1 + a \cdot (i - 2)],$$

где, например, при объединении трёх подмножеств рёбер

$\{\Gamma x_i \cup \Gamma x_j\} \cup \{\Gamma x_k\}$ $a_k = |\Gamma x_k'| / |\Gamma x_k|$, $\Gamma x_k'$ – множество рёбер, принадлежащих вершине x_k и не принадлежащих вершинам множества $\{x_i, x_j\}$.

Считая $|X_l^K| = n_l$, получим:

$$N_{3.1} = \rho^2 [n_l + a \cdot n_l \cdot (n_l - 3) / 2].$$

Поскольку $0 \leq a < 1$, $N_{3.1} < \rho^2 \cdot (n_l^2 - n_l) / 2$.

Упорядоченные множества. При i -м объединении упорядоченного множества Γx_i мощность множества, полученного на предыдущем шаге, равна $\rho + a \cdot \rho \cdot (i - 1)$. Поскольку $|\Gamma x_i| = \rho$, то используя (2) из [5], запишем:

$$N_{3.1 \text{ уп}} = 2\rho \cdot \sum_{i=1}^{n_l} [(2 + a \cdot (i - 2)) - 1],$$

Отсюда $N_{3.1 \text{ уп}} < \rho \cdot (n_l^2 + n_l) - n_l$.

Нетрудно видеть, что отношение $K_{3.1} = N_{3.1} / N_{3.1 \text{ уп}}$ при росте n_l стремится к $\rho/2$.

3.2. Определение множества $U_l^{\text{K int}} = \{u_j \in U_l^{\text{K}} : \Gamma u_j \subseteq X_l^{\text{K}} / \Gamma u_j \in \Gamma U\}$.

Неупорядоченные множества. Обозначив $|U_l^{\text{K}}| = m_l$ и считая $m_l = |U| / L$, получим количество операций сравнения, необходимых для определения множества $U_l^{\text{K int}}$:

$$N_{3.2} = A \cdot n_l \cdot m_l.$$

Упорядоченные множества. На основании (1) из [5] — $N_{3.2 \text{ уп}} = 2n_l \cdot m_l$.

Отсюда отношение $K_{3.2} = N_{3.2} / N_{3.2 \text{ уп}}$ равно $A/2$.

3.3. Определение множества $U_l^{\text{K ext}} = U_l^{\text{K}} \setminus U_l^{\text{K int}}$.

Неупорядоченные множества. Считая $|U_l^{\text{K int}}| = |U_l^{\text{K}}|/2$, имеем: $N_{3.3} = m_l^2/2$.

Упорядоченные множества. $N_{3.3 \text{ уп}} = 2(m_l + m_l/2 - 1)$.

При росте m_l отношение $K_{3.3} = N_{3.3} / N_{3.3 \text{ уп}}$ стремится к m_l .

3.4. Определение множества $\Gamma X_l^{\text{K}} = \{\Gamma x_i \in \Gamma X / x_i \in X_l^{\text{K}}\}$. Как видно из выражения при формировании ΓX_l^{K} , даже если учитывать операции записи, вычислительная сложность не зависит от упорядоченности множеств.

3.5. Определение множества $\Gamma U_l^{\text{K}} = \{\Gamma u_j : u_j \in U_l^{\text{K int}} \vee \Gamma u_j \cap X_l^{\text{K}} : u_j \in U_l^{\text{K ext}} / u_j \in U_l^{\text{K}}, \Gamma u_j \in \Gamma U\}$. Операция $\Gamma u_j \cap X_l^{\text{K}}$ будет выполняться $|U_l^{\text{K ext}}| = |U_l^{\text{K}}|/2$ раз. Не будем учитывать проверки $u_j \in U_l^{\text{K int}}$ и $u_j \in U_l^{\text{K ext}}$.

Неупорядоченные множества: $N_{3.5} = A \cdot n_l \cdot m_l/2$.

Упорядоченные множества: $N_{3.5 \text{ уп}} = m_l/2 [2(n_l + A - 1)] = m_l \cdot (n_l + A - 1)$.

При росте n_l и m_l отношение $K_{3.5} = N_{3.5} / N_{3.5 \text{ уп}}$ стремится к $A/2$.

4. Удаление куска $H_l^k(X_l^k, U_l^k)$ из модели системы. Для простоты символики будем рассматривать выполнение этой операции после первого применения алгоритма разрезания. Удаление выделенной части схемы реализуется операцией «Дополнение части до ультраграфа H_U и гиперграфа H » [6] в варианте $H(X, U) \setminus H_1^k(X_1^k, U_1^k)$. Обозначим получаемый кусок гиперграфа как $H_{сл} (X_{сл}^k, U_{сл}^k)$.

4.1. Определение $X_{сл}^k = X \setminus X_1^k$.

Неупорядоченные множества: $N_{4.1} = n \cdot n_l$.

Упорядоченные множества: $N_{4.1 \text{ уп}} = 2 (n_l + n - 1)$.

Отношение $K_{4.1} = N_{4.1} / N_{4.1 \text{ уп}}$ стремится к n_l . Напомним, что $n_l = n / L$, где L – константа.

4.2. Определение $U_{сл \text{ int}}^k = U \setminus U_1^k$.

Неупорядоченные множества: $N_{4.2} = m \cdot m_l$.

Упорядоченные множества: $N_{4.2 \text{ уп}} = 2 \cdot (m_l + m - 1)$.

Отношение $K_{4.2} = N_{4.2} / N_{4.2 \text{ уп}}$ стремится к m_l . Отметим, что $m_l = m / L$.

4.3, 4.4 и 4.5. Определение $U_{сл \text{ ext}}^k = U_1^k \text{ ext}$, $U_{сл}^k = U_{сл \text{ int}}^k \cdot U_{сл \text{ ext}}^k$ и $\Gamma X_{сл}^k = \{\Gamma x_i \in \Gamma X / x_i \in X_{сл}^k\}$. Очевидно, что на вычислительную сложность выполнения этих операций упорядочивание множеств не влияет.

4.6. Определение $\Gamma U_{сл}^k = \{\Gamma u_j : u_j \in U_{сл \text{ int}}^k \vee \Gamma u_j \setminus X_1^k : u_j \in U_{сл \text{ ext}}^k / u_j \in U_{сл}^k, \Gamma u_j \in \Gamma U\}$. Количество операций сравнения определяется выражением $|X_1^k| \cdot |\Gamma u_j| \cdot |U_{сл \text{ ext}}^k|$. Будем считать, что $|U_{сл \text{ ext}}^k| = |U_{сл}^k| / 2$.

Неупорядоченные множества: $N_{4.6} = n_l \cdot A \cdot m_l \cdot (L - 1) / 2$.

Упорядоченные множества: $N_{4.6 \text{ уп}} = (n_l + A - 1) \cdot m_l \cdot (L - 1)$.

При росте n_l и m_l отношение $K_{4.6} = N_{4.6} / N_{4.6 \text{ уп}}$ стремится к $A / 2$.

6. Проверка условий

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

Эти условия логически эквивалентны условию $X = Y$, где $Y = \{ X_1 \cdot X_2 \cdot X_3 \cdot \dots \cdot X_L \}$, проверка которого требует меньшего количества операций сравнения.

Для неупорядоченных и упорядоченных множеств соответственно имеем:

$$N_6 = n^2 \text{ и } N_{6 \text{ уп}} = 4(n-1).$$

С учётом необходимости упорядочивания множества Y получаем:

$$K_6 = N_6 / N_{6 \text{ уп}} = n^2 / (4(n-1) + n \log_2 n) = n / \log_2 n.$$

Проанализировав полученные результаты, можно сделать вывод, что наиболее существенный вклад в снижение вычислительной сложности описанного процесса декомпозиции за счёт использования операций с упорядоченными множествами вносят:

- формирование куска $H_l^k(X_l^k, U_l^k)$ гиперграфа в п. 3.3 в m_l раз;
- удаление куска $H_l^k(X_l^k, U_l^k)$ из гиперграфа в п. 4.1 в n_l и в п. 4.2 в m_l раз;
- проверка условия $X = \{ X_1 \cdot X_2 \cdot X_3 \cdot \dots \cdot X_L \}$ в $n / \log_2 n$ раз.

Выполненные исследования показывают высокую эффективность использования операций над упорядоченными множествами в алгоритмах структурного синтеза.

Литература

1. Савельев А. Я., Овчинников В.А. **Конструирование ЭВМ и систем:** Учебник для технич. вузов по спец. «Электрон. выч. маш.» –М.: Высш. шк., 1984 – 248 с.
2. 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., 1989 – 312 с.
3. Овчинников В. А. **Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем:** Учеб. для вузов.– М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. – 288 с.
4. 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.

5. Овчинников В.А. **Операции над упорядоченными множествами:** Наука и образование. Инженерное образование: Эл. Науч. Издание. – 2011. – № 6.

6. 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.

Evaluation the effectiveness of ordered set operations

77-30569/239406

10, October 2011

Ovchinnikov V.A., Ivanova G., S.

Bauman Moscow State Technical University

gsivanova@gmail.com

The efficiency of using ordered set operations is considered by example of partitioning a complex system structure into subsystems. The article considers a formal definition of this problem as separation of the hypergraph vertex set into disjoint subsets. The article describes the main design routines of computer scheme partitioning by sequential algorithm. The partitioning process was fulfilled by using unordered and ordered set operations. The computation complexity assessment of both member functions was fulfilled. The efficiency of using ordered set operations may be proportional to the quantity of network elements.

Publications with keywords: [computation complexities](#), [ordered set operations](#), [hypergraph vertex set partitioning](#)

Publications with words: [computation complexities](#), [ordered set operations](#), [hypergraph vertex set partitioning](#)

Reference

1. Savel'ev A. Ia., Ovchinnikov V.A Designing of the COMPUTER and systems, Moscow, Vysshaia shkola, 1984, 248 p.
2. Savel'ev A. Ia., Ovchinnikov V.A., Designing of the COMPUTER and systems, Moscow, Vysshaia shkola, 1984, 312 p.
3. Ovchinnikov V. A., Moscow,, BMSTU Press, 2001, 288 p.
4. Ovchinnikov V.A., Nauka i obrazovanie – Science and Education 11 (2009).
5. Ovchinnikov V.A., Nauka i obrazovanie – Science and Education 6 (2011).
6. Ovchinnikov V.A., Nauka i obrazovanie – Science and Education 12 (2009).