

Gaiduk Anatoly Romanovich – Southern Federal University; e-mail: gaiduk_2003@mail.ru; 26, Slesarnaya street, app. 2, Taganrog, 347904, Russia; phone: +78634626287; the department of automatic control systems; dr. of eng. sc.; professor.

Gelozhe Yuriy Andreevich – 44, Nekrasovskiy, GSP-17A, Taganrog, 347928, Russia; the department of radio Engineering and telecommunication systems; cand. of eng. sc.; professor.

Semenova Alexandra Vladimirovna – Ltd. Co. «DATAMICRO». e-mail: alexaforum@rambler.ru; 12, Vishnevaya street, Taganrog, 347900, Russia; phone: +78634310990; systems engineer.

УДК 681.3.06: 681.323 (519.6)

Я.Е. Ромм, И.И. Стаховская

ПАРАЛЛЕЛЬНО-КОНВЕЙЕРНОЕ УПОРЯДОЧЕНИЕ НЕОГРАНИЧЕННОГО МАССИВА

Представлены параллельно-конвейерные алгоритмы слияния с неограниченным массивом, которые построены на основе матриц сравнения. Многопутевое слияние по матрицам сравнения максимально параллельно и выполняется с единичной временной сложностью. Параллельная сортировка на этой основе выполняется с длительностью $\log_M N$ последовательных сравнений, где M – число путей слияния, N – длина последовательности. С применением многопутевого слияния синтезирован алгоритм параллельно-конвейерной сортировки массива неограниченной длины. Конвейер в каждом такте единичной длительности принимает на вход новый неупорядоченный массив из $M \times n$ элементов или новую часть такой же длины неограниченного входного массива. Время загрузки конвейера имеет единичную длительность, на выходе в такте той же длительности к упорядоченному массиву неограниченной длины добавляются новые $M \times n$ элементов с сохранением порядка всего выходного массива в целом. Даны оценки временной сложности, числа процессорных элементов, количества сегментов конвейера. Сегменты конвейера для неограниченного возрастания выходного массива образуют последовательность процессоров, количество которых на i -м шаге загрузки в сумме составит $\sum_{\ell=0}^i R_{\ell} = (2^{i+1} - 1)M^2 n^2$.

Параллельно-конвейерные алгоритмы; массив неограниченной длины; параллельная сортировка; многопутевое слияние; матрица сравнения; временная сложность.

Ya.E. Romm., I.I. Stakhovskaya

PARALLEL-CONVEYOR ORDERING ARRAY OF UNLIMITED LENGTH

Presented parallel-conveyor algorithms merger with unlimited array that is built on the basis of the comparison matrix. Multipath synthesized by comparison matrix maximum parallel and performed with the unit time complexity. Parallel sorting on this basis performed duration $\log_M N$ consecutive comparisons, where M – number of paths merger, N – consecution length. With the application of multipath synthesized fusion algorithm parallel-conveyor sorting an array of unlimited length. Conveyor unit in each cycle duration takes a new unlimited array $m \times n$ of items, or a new part of the same unlimited length of the input array. Time loading has unit length of conveyor, output in the same clock cycle continuation to an ordered array of unlimited length adding $m \times n$ new elements while the same order as a whole of the output array. Are given estimates for the time complexity, amount of processing elements, amount of conveyor seg-

ments. Conveyor segments for an unlimited increase in the output array form a consecutive of processors, amount on i -th step load for a total of $\sum_{\ell=0}^i R_{\ell} = (2^{i+1} - 1)M^2 n^2$.

Parallel-conveyor algorithm; an array of unlimited length; parallel sorting; multipath merger; matrix comparisons; time complexity.

1. Постановка вопроса. Ставится задача синтеза и анализа параллельно-конвейерных алгоритмов слияния с неограниченным массивом данных на основе матриц сравнения (МС). Актуальность этой задачи обусловлена необходимостью обработки и поиска растущих объемов информации в режиме реального времени. Понятие МС, исходные описания основанных на них параллельных алгоритмов слияния и сортировок заимствованы из [1–3]. Метод упорядочивания массива неограниченной длины строится на основе устойчивой сортировки многопутевым слиянием, выполняемой максимально параллельно с использованием МС. При этом метод состоит из двух частей. Первая – упорядочение входной информации (неупорядоченного массива) с помощью параллельной сортировки многопутевым слиянием, вторая – параллельно-конвейерное добавление к упорядоченной части массива нового упорядоченного массива путем слияния по МС. Описание составляют формализованные алгоритмы, для которых даны оценки временной сложности и числа процессоров на модели неветвящихся параллельных программ без учета обмена.

2. Параллельная сортировка многопутевым слиянием по МС. Традиционно сортировка многопутевым слиянием рассматривается как один из подходов к повышению эффективности сортировок посредством вовлечения в процесс дополнительных каналов обмена данными [4]. Ниже традиционное понятие видоизменяется следующим образом. Многопутевое слияние (M -слияние) будет применяться к M одномерным массивам, упорядоченным по отношению \leq (по неубыванию, другой порядок не рассматривается), преобразуя их в единый упорядоченный массив, слияние при этом должно сохранять порядок равных элементов. Пусть рассматривается массив

$$\{d_i\}_{i=1}^N, \quad N < \infty. \quad (1)$$

Вводятся вспомогательные определения. Одномерный массив из m упорядоченных элементов будет называться отрезком длины m . Если два отрезка

$$\{a_j\}_{j=1}^m, \quad \{b_i\}_{i=1}^n \quad (2)$$

сливаются в единый отрезок длины $m+n$

$$\{c_{\ell}\}_{\ell=1}^{m+n}, \quad (3)$$

то каждый из отрезков (2) называется сливаемым, отрезок (3) – слитым. При описании сортировки M -слиянием массива (1) будет предполагаться, если не оговорено иное, что N – целая степень постоянного числа M , $M \geq 2$. МС для 2-слияния отрезков (2) составляется из элементов

$$\alpha_{ij} = \text{sign}(a_j - b_i) = \begin{cases} 1, & b_i < a_j, \\ 0, & b_i = a_j, \\ -1, & a_j < b_i. \end{cases} \quad i=1,2,\dots,m; j=1,2,\dots,n. \quad (4)$$

При этом сливаемые отрезки дополняются ограничителями: $a_0 = b_0 = -\infty$, $a_{m+1} = b_{n+1} = \infty$ и соответственными элементами МС.

Пример. Для 2-слияния отрезков (1, 2), (2, 5) ($m=n$) набор МС имеет вид:

	$-\infty$	1	2	∞
$-\infty$				
1				
2				
∞				

МС⁽¹¹⁾

	$-\infty$	2	5	∞
$-\infty$	0	+	+	+
1	-	+	+	+
2	-	0	+	+
∞	-	-	-	0

МС⁽¹²⁾

	$-\infty$	1	2	∞
$-\infty$	0	+	+	+
2	-	-	0	+
5	-	-	-	+
∞	-	-	-	0

МС⁽²¹⁾

	$-\infty$	2	5	∞
$-\infty$				
2				
5				
∞				

МС⁽²²⁾

На основе данного набора МС определяется номер любого элемента в выходном массиве $\{c_\ell\}_{\ell=1}^{m+n}$ по следующему правилу вставки. Элемент i -й строки МС получает в выходном массиве номер, равный $i + j$, где j – номер столбца МС, в котором происходит смена знака в той же строке с неотрицательного на положительный, если МС расположена до пустой диагональной МС, и с отрицательного на неотрицательный, если МС расположена после пустой диагональной МС, в порядке слева направо. В данном примере для 2-слияния получим:

$$b_1=1 \rightarrow c_{1+0} = c_1, \quad b_2=2 \rightarrow c_{2+0} = c_2, \quad a_1=2 \rightarrow c_{1+2} = c_3, \\ a_2=5 \rightarrow c_{2+2} = c_4.$$

В общем случае количество МС линейно "по горизонтали" повторяется столько раз, сколько сливаемых отрезков, и столько же раз такая линейка повторяется по вертикали, их количество равно M^2 , размер каждой МС – $m \times n$. Ниже предполагается $m = n$. Правило вставки для общего случая имеет вид:

$$k_\ell = \ell + j_1 + j_2 + \dots + j_{i-1} + j_{i+1} + j_{i+2} + \dots + j_n, \quad (5)$$

где ℓ – номер строки МС (собственный номер элемента в сливаемом отрезке), j_r – номер столбца МС, элемент ℓ -й строки которого образует смену знака в паре с элементом $(j_r + 1)$ -го столбца. Смена знака в МС определяется как пара элементов строки вида

$$\left. \begin{aligned} \alpha_{\ell k}^{(ij)} < 0, \quad \alpha_{\ell(k+1)}^{(ij)} \geq 0, \quad \text{если } j > i, \\ \alpha_{\ell k}^{(ij)} \leq 0, \quad \alpha_{\ell(k+1)}^{(ij)} > 0, \quad \text{если } j < i, \end{aligned} \right\} \quad (6)$$

где i, j – индексы строк и столбцов, состоящих из матриц, в наборе из M^2 МС; $k = j_r, n+1 \geq \ell \geq 0 \leq k \leq n+1, 1 \leq i \leq M \geq j \geq 1$.

Разновидности правила даны в [3]. Выходной номер можно определять одновременно для всех элементов всех сливаемых отрезков, вставки взаимно независимы и могут выполняться параллельно, что всюду предполагается в дальнейшем. На основе (5), (6) в результате M -слияния сохраняется исходный взаимный порядок равных элементов, включая случай, когда они принадлежат различным отрезкам из (7). Таким образом, справедлива следующая лемма.

Лемма 1. В силу взаимной независимости всех сравнений M -слияние по МС максимально параллельно и выполняется с временной сложностью

$$t(R) \approx \tau, \quad (7)$$

где число процессоров $R = M^2 \times n^2$.

Алгоритм параллельной сортировки на основе M -слияния [3, 5]:

Шаг 1. Массив с сохранением порядка элементов разбивается на $\frac{N}{M}$ групп по M элементов. Внутри каждой группы по всем элементам выполняется M -слияние. Роль сливаемых отрезков длины $n=1$ играют элементы группы. Результат шага – $\frac{N}{M}$ отрезков, каждый длины $M \times n = M$, располагаемых слева направо в порядке исходного взаимного расположения групп.

Шаг $i \geq 2$. Пусть в результате $(i-1)$ -го шага получены $\frac{N}{M^{i-1}}$ слитых отрезков, каждый длины $n = M^{i-1}$. Эти отрезки в порядке их расположения слева направо делятся на $\frac{N}{M^i}$ групп. По $\frac{N}{M^i}$ группам и по всем элементам групп выполняется параллельное M -слияние отрезков в каждой группе. Результат шага – $\frac{N}{M^i}$ слитых отрезков длины $M \times n = M^i$, располагаемых слева направо в порядке взаимного расположения на данном шаге групп сливаемых отрезков.

На шаге $i = \log_M N$ сортировка будет закончена. Количество процессоров на i -м шаге сортировки $R = \frac{1}{2} \left(\frac{M-1}{M} \right) N M^i$.

Параллельная сортировка подсчетом на основе МС является частным случаем сортировки M -слиянием при $M = N$, $i = \log_N N = 1$, $n = N^{i-1} = 1$.

Отсюда вытекает

Теорема 1 [3]. Без учета длительности обмена параллельная сортировка M -слиянием массива из N элементов, N – целая степень числа M , $M \geq 2$, на

$R = \left(\frac{M-1}{2M} \right) N^2$ процессорах выполнима с длительностью $\log_M N$ последователь-

ных сравнений: $T \left(\left(\frac{M-1}{2M} \right) N^2 \right) \approx \log_M N \tau = O(\log_M N)$. Сортировка выполняется

по правилам вставки, инвариантным относительно M . Как частный случай при $M = N$ получается максимально параллельная форма сортировки подсчетом:

$$T \left(\frac{N^2}{2} \right) \approx \tau = O(1).$$

3. Параллельное слияние с неограниченным массивом. В [5] изложен вариант слияния упорядоченного одномерного массива переменной длины с отрезком фиксированной длины, при котором одномерный массив преобразуется в матрицу, слияние с ним выполняется последовательно по строкам, но параллельно в каждой строке, на выходе получается упорядоченный массив переменной длины. Этот процесс допускает конвейеризацию по строкам исходной матрицы [5, 6]. Обоснование алгоритма параллельной сортировки на основе M -слияния и несколько его разновидностей изложены в [6]. Ниже излагается аналог такого подхода, основанный на M -слиянии. Будем рассматривать такую организацию слияний, при которой на вход конвейера поступает одновременно M уже упорядоченных отрезков длины n , каждый предназначен для слияния с исходной матрицей (упорядоченным массивом неограниченной длины), на выходе получается единый упорядоченный массив, который интерпретируется как матрица из упорядоченных строк. Более точно, над поступившими на вход отрезками выполняется M -слияние за время $\approx \tau$ на $M^2 n^2$ процессоров согласно (9). Полученный отрезок длины Mn сливается по одной MC с первыми Mn элементами исходного массива, образуя в результате отрезок длины $2Mn$. Для выполнения этой операции требуется также $M^2 n^2$ процессоров, время слияния $\approx \tau = O(1)$. За время выполнения данного слияния на вход конвейера могут поступить новые M уже упорядоченных отрезков длины n и над ними также может быть выполнена операция M -слияния. Ее результат, в свою очередь, может передаваться по конвейеру для слияния с только что преобразованным исходным массивом, в то время как результат предыдущего преобразования продолжит слияние по одной MC со следующей частью исходного массива длины $2Mn$, на выходе получится отрезок длины $4Mn$ и т.д.

Конвейер принимает на вход в каждом такте единичной длительности новые M отрезков длины n .

В результате, временную сложность конвейера характеризует теорема.

Теорема 2. На вход конвейера в каждом такте единичной длительности поступают новые M отрезков длины n , на выходе образуется упорядоченный массив неограниченной длины. После загрузки производительность конвейера измеряется слиянием Mn элементов в одном такте длительности $\approx \tau = O(1)$. Число процессоров для слияния входных отрезков составляет $R = M^2 n^2$. Сегменты конвейера для неограниченного возрастания выходного массива образуют последовательность из $R_i = 2^i M^2 n^2$ процессоров на i -м шаге загрузки, что в сумме составит

$$\sum_{\ell=0}^i R_{\ell} = (2^{i+1} - 1) M^2 n^2 \text{ процессоров.}$$

Следствие 4. Конвейер, представленный теоремой 2, в M раз производительней конвейера на основе слияния упорядоченного одномерного массива переменной длины с отрезком фиксированной длины [5, 6] за счет того, что с интервалом τ на его вход поступают одновременно M новых отрезков длины n каждый.

4. Параллельная сортировка неограниченного массива. Если M новых отрезков длины n каждый, которые синхронно поступают на вход конвейера, не считать априори заданными, а каждый из них получать в результате сортировки, то дальнейшая организация конвейера не изменится. По длительности такта она также не изменится, если на входе упорядочение каждого отрезка производить за такт конвейера, причем синхронно по всем отрезкам. Требуемое упорядочение отрезка можно выполнить с помощью параллельной сортировки подсчетом, опи-

санной выше в качестве частного случая параллельной сортировки M -слиянием при $M = n$. Такая сортировка интерпретируется как максимально параллельная со свойством устойчивости [7] (сохраняет порядок равных элементов, кроме того, эта сортировка обладает свойством взаимно-однозначного соответствия входных и выходных индексов в явной форме [1, 8]), ее временная сложность в данном случае оценивается из соотношения $T\left(\frac{n^2}{2}\right) \approx \tau = O(1)$. Последняя оценка в точности

соответствует длительности такта рассматриваемого конвейера. Если таким образом синхронно сортировать все одновременно поступающие на вход конвейера M новых отрезков длины n , каждый из которых априори не упорядочен, то такт конвейера сохранится, равно как и его дальнейшая организация. Увеличится только оборудование для предварительного выполнения данной сортировки, оно составит $M \times \frac{n^2}{2}$ процессоров – по числу одновременно сортируемых отрезков. Таким образом, полное количество процессоров модифицированного конвейера с учетом оценки теоремы 2 на i -м шаге загрузки составит

$$R = M \times \frac{n^2}{2} + \sum_{\ell=0}^i R_{\ell} = \left(2^{i+1} - \frac{1}{2}\right) M^2 n^2. \quad (10)$$

Замечание. Изложенная организация конвейера автоматически приводит к сортировке массива неограниченной длины: достаточно заметить, что входной неупорядоченный одномерный массив длины $M \times n$ можно разделить на M отдельных отрезков длины n , которые в дальнейшем обрабатываются только что изложенным выше образом.

Теорема 3. Параллельно-конвейерная сортировка массива неограниченной длины может выполняться путем поступления в каждом такте нового неупорядоченного одномерного массива длины Mn , который разделяется на M равных отрезков, синхронно сортируемых с помощью максимально параллельной сортировки подсчетом каждый. Число процессоров для сортировки M входных отрезков длины n составит $M \times \frac{n^2}{2}$. Суммарная оценка временной сложности

данной предварительной сортировки имеет вид $T\left(M \frac{n^2}{2}\right) \approx \tau = O(1)$. На вход кон-

вейера в каждом такте единичной длительности поступают таким образом упорядоченные M отрезков длины n , на выходе конвейера образуется упорядоченный массив неограниченной длины. После загрузки производительность конвейера измеряется сортировкой Mn неупорядоченных элементов одномерного массива в одном такте длительности $\approx \tau = O(1)$. Сегменты конвейера для неограниченного возрастания выходного массива образуют последовательность из $R_i = 2^i M^2 n^2$ процессоров на i -м шаге загрузки, что в сумме с входными процессорами для предварительной сортировки составит количество R из (10).

Следствие 5. Конвейер, представленный теоремой 3, выполняет сортировку неограниченного входного массива, из которого на вход в каждом такте вырезается новая часть длины Mn .

Таким образом, в работе изложены параллельно-конвейерные схемы алгоритмов сортировки и слияния с интенсивной обработкой сверхбольших объёмов данных, что актуально в аспекте сверхбыстродействующей обработки информации [9–11].

Выводы. Представлено решение задачи синтеза и анализа параллельно-конвейерных алгоритмов слияния с неограниченным массивом данных на основе МС и аналогичных алгоритмов сортировки неограниченных массивов. Временная сложность предложенных алгоритмов оценивается на модели неветвящихся параллельных программ. Базовые алгоритмы опираются на параллельную сортировку и параллельно-конвейерное многопутевое слияние по МС. Принципиально, что конвейер в каждом такте единичной длительности принимает на вход новый неупорядоченный массив из $M \times n$ элементов или новую часть такой же длины неограниченного входного массива. Время загрузки конвейера имеет единичную длительность, при этом на выходе в каждом такте к упорядоченному массиву неограниченной длины добавляются новые $m \times n$ элементов с сохранением порядка всего

выходного массива в целом. На i -м шаге загрузки задействуются $\left(2^{i+1} - \frac{1}{2}\right)M^2 n^2$ процессоров по сегментам конвейера.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ромм Я.Е. Параллельная сортировка слиянием по матрицам сравнений. I // Кибернетика и системный анализ. – 1994. – № 5. – С. 3-23.
2. Ромм Я.Е. Параллельная сортировка слиянием по матрицам сравнений. II // Кибернетика и системный анализ. – 1995. – № 4. – С. 13-37.
3. Ромм Л.Я. Целочисленная идентификация плоских изображений с учетом множества внутриконтурных точек на основе экстремальных признаков и алгоритмов сортировки: Автореф. дис. ... канд. техн. наук. – Таганрог: ЮФУ. – 2013. – 21 с.
4. Царев Р.Ю. Структуры и алгоритмы обработки данных. – Красноярск: Сиб. федер. ун-т, 2013. – 160 с.
5. Ромм Я.Е. Бесконфликтные и устойчивые методы детерминированной параллельной обработки: Дис. ... д-ра техн. наук. – Таганрог: ТРТУ. – 1998. – 546 с.
6. Ромм Я.Е. Стаховская И.И. Параллельно-конвейерное упорядочение массива неограниченной длины. – Таганрог. гос. пед. ин-т., 2013. – 27 с.
7. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989. – 360 с.
8. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки. I // Кибернетика и системный анализ. – 2007. – № 1. – С. 165-183.
9. Гречко В.О., Фальфушинский В.В. Параллельно-конвейерные схемы алгоритмов сортировки // Труды Международной конференции «Высокопродуктивные вычисления» НРС-UA'2011. – С. 145-149.
10. Легалов А.И. Методы сортировки, полученные из анализа максимально параллельной программы // Распределенные и кластерные вычисления. Избранные материалы Третьей школы семинара. – Красноярск: Институт вычислительного моделирования СО РАН, 2004. – С. 119-134.
11. Мирошниченко С.Ю., Титов В.С. Параллельно-конвейерное устройство для векторизации аэрокосмических изображений земной поверхности // Приборостроение. – 2009. – № 2. – С. 45-52.

Статью рекомендовал к опубликованию д.т.н., профессор Л.П. Фельдман.

Ромм Яков Евсеевич – Таганрогский государственный педагогический институт; e-mail: romm@list.ru; 347926, г. Таганрог, ул. Инициативная, 48; тел.: 88634601753, 88634601812, 88634601807; кафедра информатики; зав. кафедрой; д.т.н.; профессор; член Европейской Академии Естествознания (EuANH).

Стаховская Ирина Илларионовна – e-mail: irihkast@gmail.ru; тел.: 89034604006; студентка магистратуры.

Romm Yakov Evseevich – Taganrog State Pedagogical Institute; e-mail: romm@list.ru; 48, Initsiativnaya street, Taganrog, 347926, Russia; phones: +78634601753, +78634601812, +78634601807; the department of information science; head of the department; dr. of eng. sc.; professor.

Stakhovskaya Irina Illarionovna – e-mail: irihkast@gmail.ru; phone: +79034604006; the student grade.

УДК 681.3.06(075)

М.И. Ледовской

АЛГОРИТМ ВЫЧИСЛЕНИЯ ФУНКЦИЙ SIN(X) И COS(X) ДЛЯ 16-РАЗРЯДНЫХ МИКРОКОНТРОЛЛЕРОВ

При использовании микроконтроллеров во встраиваемых системах различного назначения возникает необходимость вычисления элементарных функций $\sin(x)$ и $\cos(x)$. Решение этой задачи с помощью стандартной библиотеки функций языка C приводит к существенному росту программного кода и времени вычислений для микроконтроллеров с ограниченными вычислительными возможностями. В этом случае альтернативой могут служить собственные подпрограммы на основе таблично-алгоритмического метода, реализуемого с помощью целочисленной арифметики микроконтроллера. Однако преимущества таблично-алгоритмического метода проявляются в том случае, если поправки к табличным значениям функций вычисляются с помощью простейших методов аппроксимации. Негативным следствием применения указанных методов является значительный объем табличных значений функций, которые требуется хранить в памяти микроконтроллера. По этой причине реализация таблично-алгоритмического метода затрудняется. В настоящей работе предлагается алгоритм таблично-алгоритмического метода, который по сравнению с аналогом обеспечивает требуемую точность и позволяет сократить в 8 раз объем табличных значений функций. Алгоритм может найти применение, например, при создании различных устройств со сверхнизким энергопотреблением на базе 16-разрядных микроконтроллеров MSP430 без аппаратного умножителя.

Функция $\sin(x)$; $\cos(x)$; таблично-алгоритмический метод; целочисленный алгоритм, микроконтроллер; погрешность; MATLAB; модель.

M.I. Ledovskoy

THE ALGORITHM OF CALCULATION OF THE FUNCTIONS SIN(X), COS(X) FOR 16-BIT MICROCONTROLLERS

When using microcontrollers and embedded systems for various purposes there is a necessity of calculation of elementary functions $\sin(x)$ and $\cos(x)$. The solution to this problem using standard C-library functions with leads to substantial growth of code and time of calculations for microcontrollers with limited computing power. In this case, the alternative can serve as your own routines based table-algorithmic method, which is implemented using integer arithmetic microcontroller. However, the advantages of table-algorithmic method appear in the case, if the amendments to the tabular values of the functions are computed using the simplest methods of approximation. A negative consequence of use of the mentioned methods is a significant amount of tabular values of the functions that you want to store in memory of the microcontroller. For this reason, the implementation of a table-algorithmic method difficult. In the present paper an algorithm is proposed table-algorithmic method, which in comparison with the similar provides the required accuracy and reduces to 8 times the amount of the tabular values of functions. The algorithm can find application, for example, the creation of different devices with ultra low power consumption-based 16-bit MSP430 microcontrollers without hardware multiplier.

The function $\sin(x)$; $\cos(x)$; table-algorithmic method; integer algorithm; microcontroller; error; MATLAB; model.