

УДК 681.513.8

ОБОБЩЁННЫЙ РЕЛАКСАЦИОННЫЙ АЛГОРИТМ МГУА

А.В. Павлов

*Международный научно-учебный центр ЮНЕСКО информационных технологий и систем
НАНУ и МОНУ, 03680, Киев, пр. Глушкова 40**Me_ovechka@bigmir.net*

Abstract. The paper gives an architecture analysis of Multilayered Simplified Algorithm (MSA) of GMDH, its advantages and disadvantages have been revealed. Generalized relaxational iterative algorithm of GMDH free from the MSA disadvantages has been introduced having an essentially new architecture for iterative GMDH algorithms that build models linear in coefficients.

Keywords: Group Method of Data Handling (GMDH), multilayered simplified algorithm, generalized relaxational iterative algorithm, algorithm architecture.

В работе осуществлён анализ архитектуры Многорядного Упрощенного Алгоритма (МУА) МГУА, выявлены его недостатки и преимущества. Предложен обобщённый релаксационный итерационный алгоритм, не содержащий недостатков МУА и имеющий принципиально новую архитектуру для итерационных алгоритмов МГУА, строящих модели, линейные по параметрам.

Ключевые слова: Метод Группового Учета Аргументов (МГУА), многорядный упрощённый алгоритм, обобщённый релаксационный итерационный алгоритм МГУА, архитектура алгоритма.

Вступление

В работе выполняется анализ архитектуры многорядного упрощённого алгоритма (МУА) МГУА [1] и предлагается обобщённый релаксационный итерационный алгоритм МГУА, свободный от недостатков МУА МГУА.

1. Постановка задачи

Пусть имеем выборку данных, заданную матрицей $W = (X:y)$, $\dim X = n_W \times m$, $\dim y = n_W \times 1$, $x_i \in \mathfrak{R}^{n_W}$, $i = \overline{1, m}$, где n_W – количество наблюдений (точек), m – число аргументов (вектор-столбцов матрицы X). Предполагается, что выходная переменная \check{y} является некоторой функцией $\check{f}(\check{X}_{\check{M}}, \check{\Theta}_{\check{M}})$ от истинных аргументов $\check{X}_{\check{M}} \subseteq X$ и неизвестного истинного вектора параметров $\check{\Theta}_{\check{M}}$, $\dim \check{X}_{\check{M}} = n_W \times \check{M}$, $\dim \check{\Theta}_{\check{M}} = \check{M} \times 1$. Множество аргументов матрицы X выбирается из условий конкретной задачи, но, кроме истинных аргументов, может содержать и фиктивные (лишние). В выходной переменной \check{y} присутствует аддитивный шум ξ , который является случайной величиной, распределённой по нормальному закону с нулевым математическим ожиданием и неизвестной конечной дисперсией:

$$y = \check{y} + \xi = \check{f}(\check{X}_{\check{M}}, \check{\Theta}_{\check{M}}) + \xi.$$

По данным наблюдений необходимо построить функцию

$$\hat{y}^* = f^*(X_{M^*}^*, \hat{\Theta}_{M^*}^*), \quad (1)$$

при которой целевая функция (критерий) принимает минимальное значение $CR^*(y, \hat{y}^*)$. Заданному критерию соответствует тройка:

$$\{f^*, X_{M^*}^*, \hat{\Theta}_{M^*}^*\} = \arg \min_{f \in \Phi, X_M \subseteq X, \Theta_M \in \mathfrak{R}^M} CR(y, \hat{y}) = \arg \min_{f \in \Phi, X_M \subseteq X, \Theta_M \in \mathfrak{R}^M} CR(y, f(X_M, \hat{\Theta}_M)),$$

где Φ – множество функций заданного класса, генерируемых соответствующим алгоритмом МГУА (примерами классов являются тригонометрические, полиномиальные и другие функции, линейные или нелинейные разностные уравнения и пр.); X_M – некоторая подматрица матрицы X , $\dim X_M = n_W \times M$, $M \leq m$; $\hat{\Theta}_M$ – вектор оценок параметров конкретной функции f , $\dim \Theta_M = M \times 1$; \mathfrak{R}^M – M -мерное пространство действительных значений вектора параметров Θ_M .

В данной постановке выделим две задачи:

1) задачу дискретной оптимизации множества аргументов матрицы X , преобразованных в заданном классе функций

$$\{f^*, X_{M^*}^*, \hat{\Theta}_{M^*}^*\} = \arg \min_{f \in \Phi, X_M \subseteq X} CR(y, \hat{y}) = \arg \min_{f \in \Phi, X_M \subseteq X} CR(y, f(X_M, \hat{\Theta}_M)); \quad (2)$$

2) задачу оптимизации непрерывных значений параметров Θ_M функции f , зависящей от множества аргументов X_M

$$\hat{\Theta}_M = \arg \min_{\Theta \in \mathfrak{R}^M} QR(y, \hat{y}) = \arg \min_{\Theta \in \mathfrak{R}^M} QR(y, f(X_M, \Theta_M)). \quad (3)$$

Структурой модели будем называть функцию $f(X_M, \Theta_M)$, заданную с точностью до значений параметров Θ_M . Полностью заданную функцию $f(X_M, \hat{\Theta}_M)$ будем называть некоторым решением или моделью, а функцию $f^*(X_{M^*}^*, \hat{\Theta}_{M^*}^*)$ – оптимальным решением задачи (оптимальной моделью). В (3) QR является критерием, в соответствии с которым определяются оптимальные значения вектора параметров $\hat{\Theta}_M$ для некоторой структуры модели; в (2) CR – критерий для определения оптимальной структуры модели и, в конечном итоге, оптимальной модели. Под построением модели понимается генерация её структуры и оценивание параметров, соответствующих этой структуре.

МУА МГУА в общем случае строит как статические, так и динамические степенные аддитивно-мультипликативные модели, включая на каждой итерации один аддитивный одночлен. В результате структура модели может иметь, например, вид:

$$y = \theta_0 + \theta_1 \frac{x_1}{x_2^2} + \theta_2 x_2^3 x_1 x_4 + \theta_3 \frac{1}{x_3 x_1^2}.$$

Мультипликативным одночленом будем называть одночлен, состоящий из произведений соответствующих элементов вектор-столбцов входной матрицы. Если в [2] анализ МУА МГУА был сконцентрирован на алгоритмах оценки

коэффициентов Θ и расчете критериев селекции CR , то в данной работе содержится подробный анализ процесса работы алгоритма в целом.

2. Анализ МУА МГУА

В архитектуре МУА МГУА можно выделить три блока (рис. 1):

- блок формирования обобщённых переменных, в котором имеется возможность увеличить размерность исходного пространства;
- блок подготовительных вычислений, результаты которых позже используются при оценивании параметров Θ и расчёта критериев селекции модели;
- блок итераций, где непосредственно строится модель.

На каждой итерации выполняется Алгоритм Поиска Одночлена (АПО), который нужно включить в модель. АПО состоит из трёх этапов:

1. Для построения F_1 лучших моделей, линейных по входным переменным, сначала строятся m моделей вида:

$$\tilde{y}_{A,r+1,i} = \hat{\omega}_{A,r+1,i}^{r+1} \tilde{x}_{A,r+1,i}, \quad i = 1, \dots, m, \quad r = 0, \quad (4)$$

$$\tilde{y}_{A,r+1,i} = \hat{\omega}_{A,r+1,i} \tilde{y}_{A,r} + \hat{\omega}_{A,r+1,i}^{r+1} \tilde{x}_{A,r+1,i}, \quad i = 1, \dots, m, \quad r > 0, \quad (5)$$

где $\tilde{y}_{A,r+1,i}$ – оценка решения для центрированного на обучающей выборке A выхода \tilde{y} , полученное на $(r+1)$ -й итерации для i -й модели; $\tilde{x}_{A,r+1,i}$ – аргумент (центрированный на выборке A столбец матрицы X), добавляемый в модель на $(r+1)$ -й итерации; $\tilde{y}_{A,r}$ – оценка лучшего решения предыдущей итерации; $\hat{\omega}_{A,r+1,i}^{r+1}$, $\hat{\omega}_{A,r+1,i}$ – оценки коэффициентов, определяемые по методу наименьших квадратов. Отбираются F_1 лучших моделей, где F_1 – свобода выбора, $F_1 \leq m$.

2. Если строятся аддитивно-мультипликативные модели, то для каждой i -й модели (4), (5), $i = 1, \dots, F_1$ осуществляется Процедура Конструирования Мультипликативных Одночленов (ПКМО).

3. Выбор по критерию CR одной лучшей ($F=1$) среди F_1 построенных моделей, где F – свобода выбора моделей на каждой итерации.

После окончания каждой итерации проверяется одно из условий останова алгоритма:

- $CR_{r+1} > CR_r$ или
- $r+1 = R$,

где R – наперед заданное количество итераций алгоритма. Если условие останова не выполняется, выбранная модель передаётся на следующую итерацию.

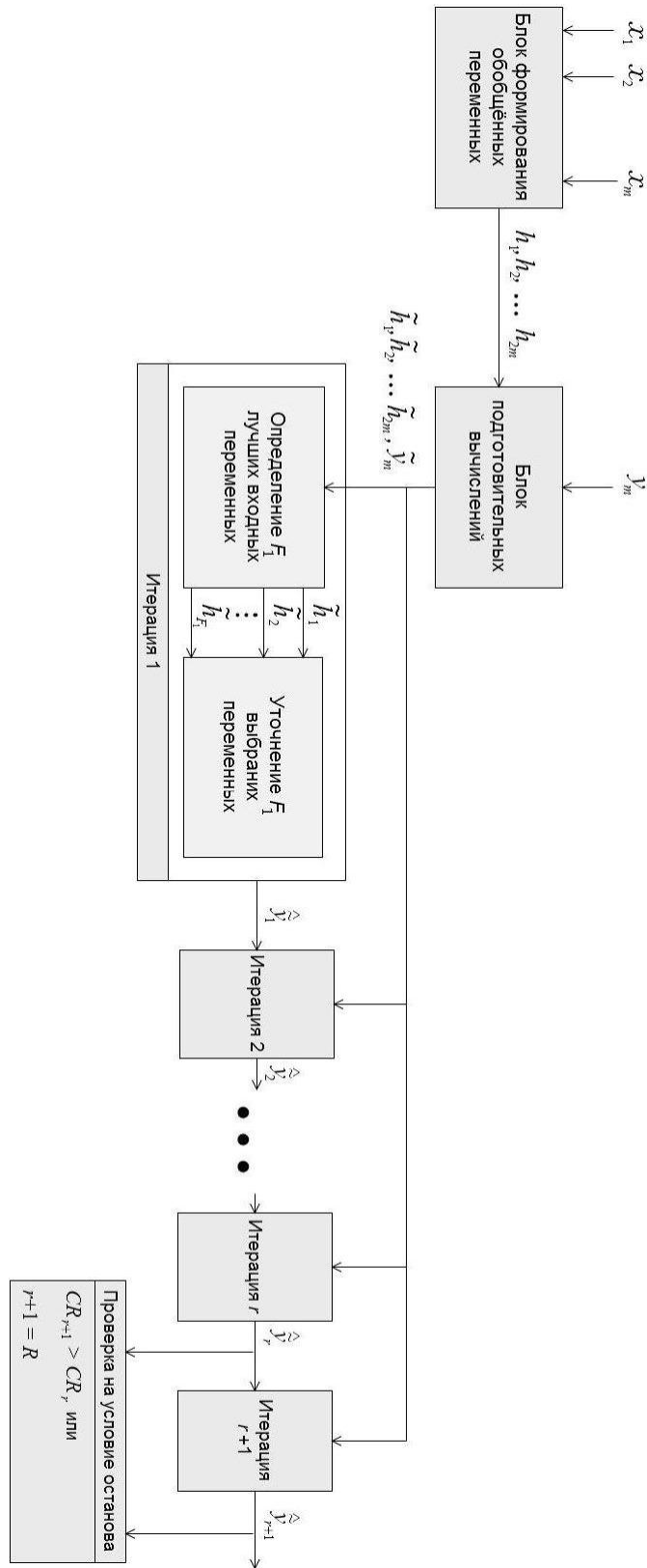


Рис. 1. Архитектура МУА МГУА и процесс его работы. Процедура конструирования мультипликативных одночленов.

Входным параметром ПКМО является *multNum* – максимальное количество мультипликаций (умножений), которые допускаются для формирования одночлена $z_{A,r+1,i}$, при исходном его значении, совпадающем с

$x_{A,r+1,i}$. Пусть it текущее количество мультипликаций; CR_{it} – значение критерия CR для лучшей модели итерации it -й мультипликации. Блок-схема ПКМО представлена на рис. 2.

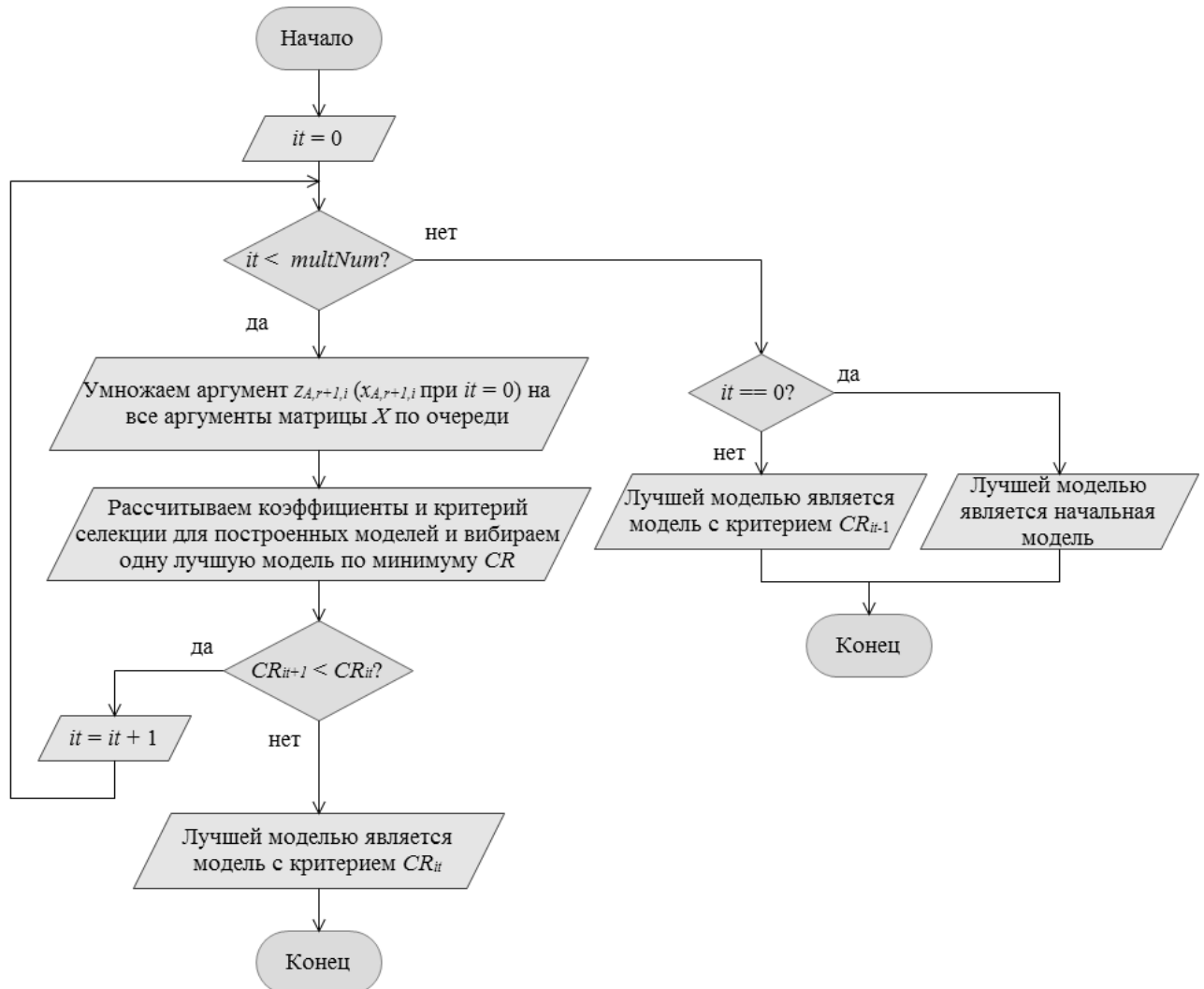


Рис. 2. Блок-схема ПКМО.

Далее при условии, что $it < multNum$ осуществляется построение моделей вида:

$$\tilde{y}_{A,r+1,i} = \omega_{A,r+1,i}^{r+1} \tilde{z}_{A,r+1,i} + \omega_{A,r+1,i} \hat{y}_{A,r}, \quad i = 1, \dots, m,$$

где элементы вектора $\tilde{z}_{A,r+1,i}$ являются центрированными на выборке A значениями вектора-столбца $z_{A,r+1,i}$. В случае $it > 0$ вектор $z_{A,r+1,i}$ обозначен как $z_{A,r+1,i,it}$, при $it = 1$ элементы вектора рассчитываются по формуле

$$z_{A,r+1,i,it,j} = x_{A,r+1,i,j} \cdot x_{A,r+1,k,j}, \quad j = 1, \dots, n_A; \quad i, k = 1, \dots, m,$$

а при $it > 1$ по формуле

$$z_{A,r+1,i,it+1,j} = z_{A,r+1,i,it,j} \cdot x_{A,r+1,k,j}, \quad j = 1, \dots, n_A; \quad i, k = 1, \dots, m.$$

Работу АПО удобно представить в виде дерева, изображенного на рис. 3. Пусть матрица аргументов $X = (x_1, x_2, x_3)$; свобода выбора лучшего мультипликативного одночлена $F_1 = 3$; $r=1$ – первая итерация; $multNum = 2$.

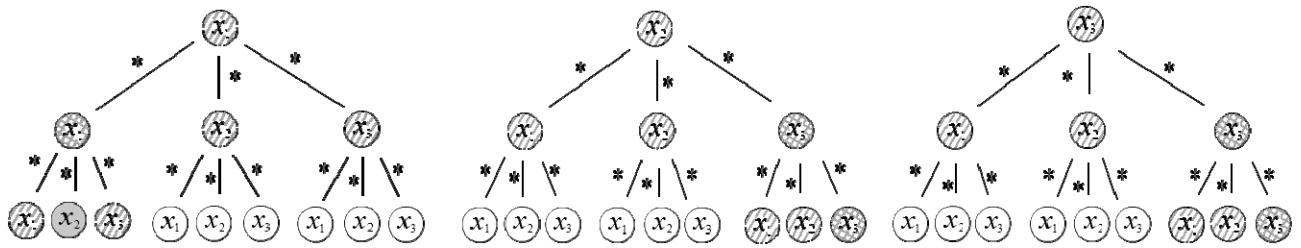


Рисунок 3. Иллюстрация работы АПО МУА в виде дерева:

«*» – знак умножения; x_i – рассчитанный одночлен; x_1 – нерассчитанный одночлен; x_2 – лучший одночлен на итерации мультипликации; x_3 – одночлен, который необходимо ввести в модель на $(r+1)$ -й итерации.

На рис. 3 виден существенный недостаток реализации АПО: при построении дерева, расчет некоторых моделей производится дважды. Следует отметить, что на текущей итерации мультипликации выбор F_1 лучших моделей осуществляется путём селекции одной лучшей модели-потомка для каждой модели-родителя, а не выбором F_1 лучших среди всех моделей-потомков. Этот факт объясняет целесообразность генерации всех возможных (m штук) моделей-потомков для каждой модели-родителя.

Также очевидно, что АПО реализует направленный перебор, в который не входят мультипликативные одночлены, присутствующие в полном переборе: $x_1 x_2 x_3, x_2^3$, т.е. в результате имеем усеченное дерево структур. Хотя при условии $multNum \leq 1$, АПО всегда реализует полный перебор.

Таким образом, при $multNum > 1$ АПО не гарантирует нахождение глобального оптимума критерия CR на текущей итерации, что в свою очередь уменьшает вероятность нахождения глобального оптимума критерия в целом. Поэтому отсутствие полного перебора в АПО можно отнести к недостаткам архитектуры МУА МГУА.

Учитывая анализ МУА МГУА [2] и сказанное выше, данный алгоритм имеет следующие недостатки:

1. Архитектура алгоритма не реализует принцип неокончательных решений ($F = 1$), что при условии неполного перебора структур является нежелательным, т.к. не гарантирует нахождение истинной структуры модели при отсутствии шума в данных.
2. АПО не гарантирует нахождение глобального оптимума критерия CR на множестве перебираемых структур.
3. Алгоритмы оценки параметров и расчёта критериев селекции модели зависят от количества наблюдений.
4. АПО выполняет больше операций, чем необходимо при построении моделей с мультипликативными одночленами, поскольку имеются повторные генерации одночленов.

К преимуществам МУА МГУА относится экономия оперативной памяти, поскольку в оперативной памяти сохраняется только матрица H , содержащая исходные и преобразованные входные переменные. Память под мультипликативные одночлены выделяется и освобождается динамически при их конструировании.

3. Обобщенный релаксационный итерационный алгоритм МГУА

В основе Обобщенного Релаксационного Итерационного Алгоритма (ОРИА) МГУА лежат некоторые идеи МУА МГУА, а именно: добавление только одного одночлена в модель на каждой итерации; центрирование аргументов с целью оценки только двух коэффициентов; алгоритм (видоизмененный) направленного перебора при поиске лучшего мультипликативного аргумента.

Мультипликативным аргументом будем называть аргумент, состоящий из произведений соответствующих элементов вектор-столбцов входной матрицы.

В результате анализа известных алгоритмов МГУА в [3] выделены следующие их составляющие:

- класс моделей;
- генератор структур моделей;
- метод оценивания параметров;
- алгоритм расчета критериев селекции.

ОРИА включает в себя три алгоритма, описанных в [4], для оценивания параметров и расчёта критериев селекции:

- алгоритм с нерекуррентными вычислениями;
- два варианта алгоритмов с рекуррентными вычислениями.

Каждый из этих алгоритмов устраняет третий недостаток МУА МГУА: зависимость от числа точек.

Кроме того, предлагаются два генератора структур:

- с направленным перебором, в основе которого лежит идея генератора МУА МГУА, однако его реализация не имеет выявленного четвёртого недостатка;
- с полным перебором мультипликативных аргументов, устраняющий второй недостаток МУА МГУА.

Таким образом, в ОРИА можно выделить три компонента:

- 1) класс моделей (алгоритм реализует следующие модели многих переменных: степенные, аддитивно-мультипликативные и их разностные аналоги);
- 2) генератор структур (с полным и направленным перебором);
- 3) алгоритмы оценки параметров и расчёта критерия селекции (алгоритм с нерекуррентными вычислениями; два варианта алгоритмов с рекуррентными вычислениями).

Каждая из трёх компонент ОРИА является независимой от остальных, т.е. можно конструировать различные РИА, взяв по одному из вариантов каждой компоненты. Отметим, что для компоненты «класс моделей» возможна комби-

нация вариантов, поскольку, например, можно построить разностно-полиномиальную модель.

На рис. 4 показаны составляющие процесса построения ОРИА-модели.



Рис. 4. Составляющие ОРИА и процесс построения модели.

В архитектуре ОРИА компонент «класс моделей» не специфицирует конкретный релаксационный итерационный алгоритм, поскольку фактически данный компонент является составляющей блока преобразования исходных данных. Имея матрицу X , и выполнив соответствующие преобразования её аргументов, можно сформировать указанный класс моделей. При этом созданная матрица H рассматривается в качестве исходной в любом возможном РИА МГУА.

Предложенную архитектуру ОРИА МГУА можно обобщить на любой алгоритм МГУА, строящий модели, линейные по параметрам Θ . Все возможные варианты алгоритмов РИА МГУА, объединенные в ОРИА МГУА, представлены на рис. 5.

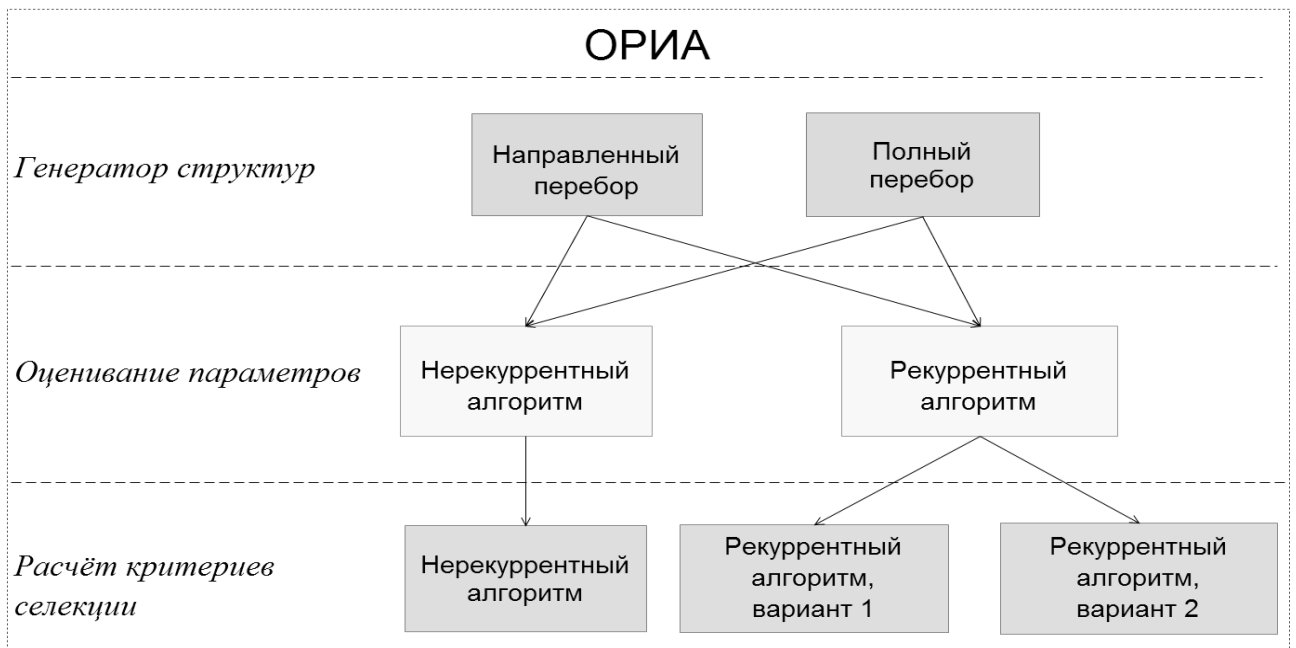


Рисунок 5. Схема получения возможных РИА в ОРИА МГУА.

Детальная схема работы ОРИА показана на рис. 6. Выделим три стадии:

1. Стадию преобразования матрицы X при формировании класса моделей.
2. Подготовительную стадию, на которой выполняются вспомогательные вычисления, использующиеся на стадии построения моделей.
3. Стадию итераций, на которой непосредственно осуществляется построение моделей.

В данной архитектуре в алгоритм вводится свобода выбора $F > 1$ лучших моделей, которые переходят на следующую итерацию, что устраняет первый недостаток МУА МГУА.

Первая стадия ОРИА состоит из последовательности двух этапов:

- 1) осуществление унарных преобразований аргументов матрицы X ;
- 2) формирование мультипликативных аргументов.

Основой для осуществления унарных преобразований является множество функциональных трансформаций Ψ , которое в текущей версии алгоритма состоит из двух элементов:

$$\Psi = \{\varphi_1(x) = 1/x, \varphi_2(x, l) = \text{lag}(x, l)\},$$

где $\text{lag}(x, l) = x_{-l}$ – функция взятия l -го запаздывания (лага) от переменной x . В дальнейшем Ψ может быть расширена, например, функциями $\ln(x)$, $\exp(x)$, $\sin(x)$ и т. д. При осуществлении преобразований имеется возможность выполнять суперпозицию функций, например, $\varphi_1(\varphi_2(x_1, 5)) = 1 / x_{1,-5}$.

После применения унарных преобразований можно осуществить k -арное преобразование (мультипликацию) над аргументами исходной матрицы с заданной максимальной степенью.

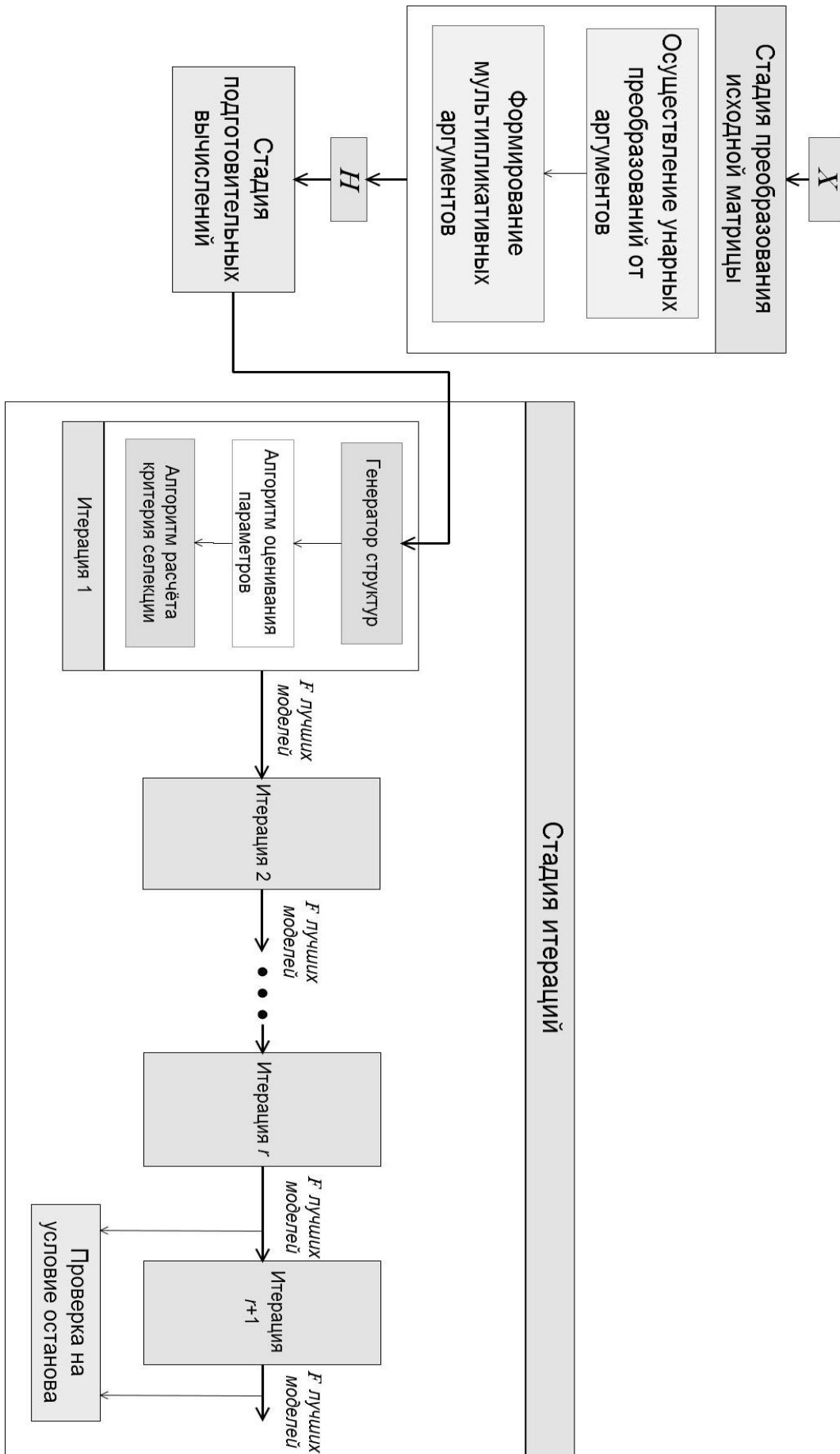


Рис. 6. Архитектура ОРИА и процесс его работы.

Опишем данную процедуру. Без ограничения общности, пусть на вход процедуры формирования мультипликативных аргументов поступает исходная матрица X . Необходимо сформировать матрицу Z , состоящую из произведений соответствующих элементов вектор-столбцов матрицы X без повторов полученных столбцов. Пусть $\dim X = n_A \times m$ и задана максимальная степень полинома pow ($pow > 1$), которую нужно получить для элементов вектор-столбцов матрицы X . Тогда элементы вектор-столбца z матрицы Z рассчитываются по формуле:

$$z_j = \prod_{k=1}^K v_{kj}, j=1, \dots, n_A,$$

где $v_k \in \{x_1, \dots, x_m\}$, $k = 1, \dots, K$, $K = 2, 3, \dots, pow$.

Расширенная матрица $H = (X : Z)$, $\dim H = n_A \times C_{m+pow}^{pow} - 1$.

На стадии подготовительных вычислений происходит центрирование вектор-столбцов матрицы $W = (H:y)$ на выборке A , а также вычисление матриц системы нормальных уравнений:

$$\tilde{H}_{A,r}^T \tilde{H}_{A,r}, \tilde{H}_{B,r}^T \tilde{H}_{B,r}, \tilde{H}_{A,r}^T \tilde{y}_{A,r}, \tilde{H}_{B,r}^T \tilde{y}_{B,r},$$

элементы которых используются при построении моделей на последней стадии ОРИА МГУА.

Условие останова ОРИА – аналогичное условию останова МУА МГУА, при этом CR_r и CR_{r+1} обозначают значения критериев селекции для самых лучших из $F > 1$ лучших моделей на r -й и $(r+1)$ -й итерации соответственно.

Опишем новую реализацию генератора направленного перебора.

4. Алгоритм поиска мультипликативного одночлена ОРИА МГУА

В результате анализа АПО МУА в основе реализации АПО ОРИА предлагается использование префиксного дерева [5]. Префиксное дерево – это абстрактная структура данных, которая позволяет сохранять и осуществлять поиск ассоциативного массива, ключами которого являются строчки. Для построения дерева задаётся алфавит символов \aleph , из которых составляются слова (строчки). Значение ключа получается путём просмотра всех родительских узлов, каждый из которых сохраняет один или несколько символов алфавита, и конкатенации их символов в порядке просмотра.

В нашем случае алфавит символов \aleph – это индексы вектор-столбцов матрицы X : Множеству индексов $\aleph = \{‘0’, ‘1’, \dots, ‘m-1’\}$ соответствует множество $\{‘x_1’, ‘x_2’, \dots, ‘x_m’\}$. Словами этого алфавита являются: ‘012’, которому соответствует $‘x_1x_2x_3’$; ‘000’ – $‘x_1x_1x_1’$ и т. д. Каждому слову ставится в соответствие вектор-столбец из произведений элементов закодированных вектор-столбцов матрицы X , например, слову ‘000’ – вектор-столбец x_1^3 .

Блок-схема алгоритма представлена на рис. 7.

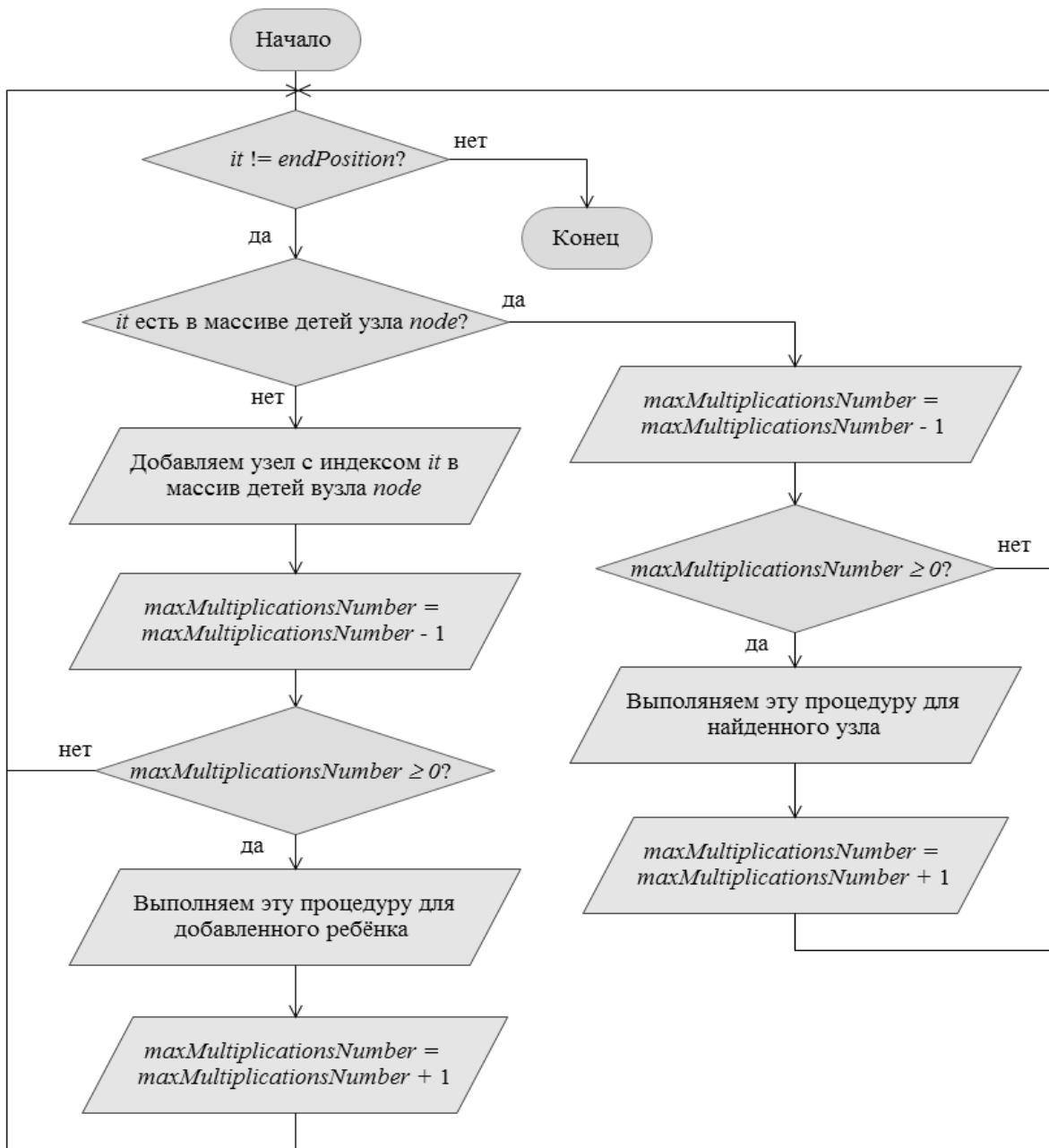


Рис. 7. Блок-схема алгоритма построения префиксного дерева.

Алгоритм построения префиксного дерева реализуется с помощью следующей процедуры:

AddChildren(node, beginPosition, endPosition, maxMultiplicationsNumber),

где: node – узел, в который добавляются потомки; beginPosition – указатель на начальный элемент массива индексов; endPosition – указатель на элемент, следующий за последним элементом массива индексов, например, null; maxMultiplicationsNumber – максимальное количество мультипликаций.

Перед запуском процедуры необходимо инициализировать указатели beginPosition и endPosition, используя исходный массив индексов символов indicesArray. Данный алгоритм строит полное дерево структур, представленное на рис. 8.

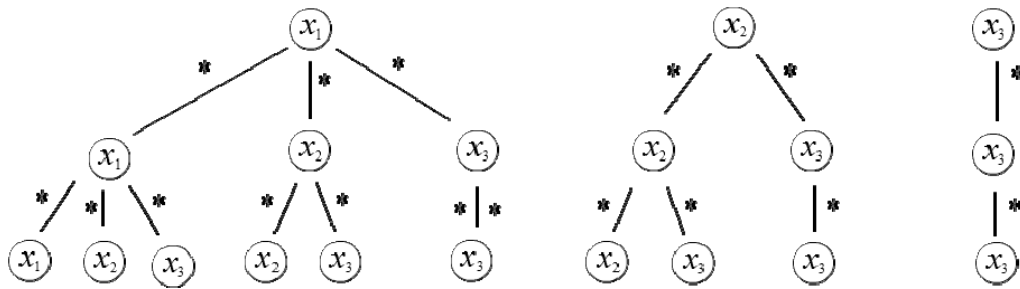


Рис. 8. Результат работы алгоритма построения префиксного дерева.

Изображенное дерево является минимальным (не имеет повторов), поэтому при использовании алгоритма обхода префиксного дерева быстроедействие ОРИА по сравнению с МУА увеличивается (см. рис. 8).

Алгоритм обхода префиксного дерева.

1. Строятся модели для узлов первого уровня и из них отбираются F_1 лучших по минимуму критерия CR . Отобранные модели добавляются в массив `bestModelsArray`.

2. Для каждого выбранного узла первого уровня выполняется процедура `CalculateNode(parentNode, parentNodeFitnessCriterion)`, где `parentNode` – узел, для потомков которого будут строиться модели; `parentNodeFitnessCriterion` – значение критерия этого узла. В нашем случае `parentNode` является текущим выбранным узлом первого уровня.

Процедура `CalculateNode`.

2.1 Строятся модели для каждого из узлов-детей родительского узла `parentNode`. Определяются среди них F_2 лучших моделей и добавляются в массив `bestModelsArray`.

2.2 Для каждого из F_2 выбранных узлов рекурсивно выполняется процедура `CalculateNode` при выполнении следующих условий:

а) значение критерия CR модели, построенной с включением нового узла меньше `parentNodeFitnessCriterion`;

б) выбранный узел имеет потомков.

2.3 Среди всех моделей массива `bestModelsArray` выбирается одна лучшая.

В отличие от реализации АПО МУА, данная реализация является более гибкой, поскольку позволяет варьировать количество лучших моделей F_2 , которые переходят на следующую it -у итерацию мультипликации при $it > 1$. При $F_2 = 1$ получаем АПО МУА.

Для создания матрицы Z на первой стадии ОРИА удобно воспользоваться процедурой построения префиксного дерева. При добавлении узла-потомка рассчитывается вектор $z_{i,it+1}$, соответствующий этому узлу по формуле:

$$\begin{cases} z_{j,i,it+1} = z_{j,i,it} \cdot x_{j,k}, & it > 1 \\ z_{j,i,1} = x_{j,i} \cdot x_{j,k}, & it = 1 \end{cases}, j = 1, \dots, n_A, k, i \in \{1, \dots, m\},$$

где it – номер итерации мультипликаций; x_i – узел верхнего уровня дерева; $z_{i,it}$ – вектор родителя.

Описанный подход имеет следующие преимущества:

1. Строится префиксное дерево, которое в дальнейшем используется в АПО.
2. Формируется матрица H .
3. Процедура имеет минимальное количество операций умножения.

5. Заключение

В работе осуществлён анализ архитектуры МУА МГУА, выявлены её недостатки и преимущества. С целью устранения недостатков предложен обобщённый релаксационный итерационный алгоритм МГУА, основными преимуществами которого являются:

- независимость от числа точек в процессе поиска структуры модели;
- архитектура, реализующая принцип неокончательных решений ($F > 1$), способствующая нахождению глобального оптимума критерия CR при использовании генератора структур с направленным перебором;
- новая реализация алгоритма поиска лучшего мультипликативного одночлена, позволяющая ускорить работу МУА МГУА;
- генератор структур с полным перебором находит оптимум критерия CR на текущей итерации алгоритма. Этот алгоритм назовём релаксационный итерационный алгоритм с полным деревом структур (РИА ПДС).

Предложена принципиально новая архитектура для алгоритмов МГУА, строящих модели, линейные по параметрам.

Литература

1. Шелудько О.И. Самоорганизация математических моделей при решении некоторых задач надежности и контроля. // Дисс. на соиск. учен. степ. канд. техн. наук. – Киев. – 1975. – 166 с.
2. Павлов А.В. Модифицированный алгоритм с комбинаторной селекцией и ортогонализацией переменных и его анализ // Індуктивне моделювання складних систем. Збірник наук. праць. – К.: МННЦІТС, – 2010. – С. 130-139.
3. Степашко В.С. Алгоритмы МГУА как основа автоматизации процесса моделирования по экспериментальным данным // Автоматика. - 1988. - № 4. - С.44-55.
4. Павлов А.В., Степашко В.С. Рекуррентные алгоритмы расчёта коэффициентов и критериев селекции в релаксационном алгоритме МГУА // Кибернетика и вычислительная техника, – 2011. – вып. 165. – С.72-82.
5. Префиксное_дерево – доступно на сайте <http://ru.wikipedia.org/wiki/>.