УДК 681.513.8

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ С КОМБИНАТОРНОЙ СЕЛЕКЦИЕЙ ПЕРЕМЕННЫХ И ЕГО АНАЛИЗ

А.В.Павлов
Международный научно-учебный центр информационных технологий и систем НАН и МОН Украины
me_ovechka@bigmir.net

В статті викладено опис модифікованого алгоритму МГУА з комбінаторною селекцією і ортогоналізацією змінних, включаючи геометричну інтерпретацію алгоритму оцінки параметрів. Доведена внутрішня збіжність оцінок коефіцієнтів моделі до оцінок МНК при нескінченному збільшенні кількості рядів алгоритму. Проведено аналіз алгоритму і виявлені його особливості, переваги та недоліки.
Ключові слова: модифікований алгоритм МГУА з комбінаторною селекцією і ортогоналізацією змінних, багаторядний алгоритм, МГУА, МАКСО

The paper gives a description of Modified Algorithm with Combinatorial-Selection of Orthogonalized factors (MACSO) and geometric interpretation its parameters estimation procedure. Internal convergence of the parameters estimation algorithm to estimations obtained using LSM has been proved. Features, advantages and drawbacks of the algorithm are revealed.
Key words: Multilayered algorithm, GMDH, MACSO, orthogonalized factor.

В статье изложено описание модифицированного алгоритма МГУА с комбинаторной селекцией и ортогонализацией переменных, включая геометрическую интерпретацию алгоритма оценки параметров. Доказана внутренняя сходимость оценок коэффициентов модели к оценкам МНК при бесконечном увеличении количества рядов. Проведён анализ алгоритма и выявлены его особенности, преимущества и недостатки.
Ключевые слова: модифицированный алгоритм МГУА с комбинаторной селекцией и ортогонализацией переменных, многорядный алгоритм, МГУА, МАКСО

Введение. Многие практические задачи не удаётся решить применением комбинаторных алгоритмов МГУА, используя современные сверхмощные вычислительных системы. Такие системы расширяют множество перебираемых аргументов на небольшое количество. Известные многорядные алгоритмы МГУА неполного перебора, позволяют строить модели с гораздо большим количеством аргументов [1]. Несмотря на то, что классический алгоритм нашёл широкое применение в решении практических задач, он имеет ряд недостатков [2]. На данный момент существует множество модификаций классического многорядного алгоритма, включая использование идей генетической селекции и клонирования [3], его гибридизации с дифференциальным эволюционным алгоритмом [4], а также идеями, позаимствованными из природы: муравьиные алгоритмы [5], алгоритмы поведения стаи рыб [6]. Необходимо упомянуть о гибридных нейронных сетях (НС), являющихся синтезом классического многорядного алгоритма и НС [7-8]. Особенностью данных алгоритмов является построение относительно точных моделей за сравнительно короткое время. И оригинал – модифицированный упрощённых алгоритм МГУА (МУА МГУА), и его усовершенствование – модифицированный алгоритм с

комбинаторной селекцией и ортогонализацией переменных (МАКСО) принадлежит классу многорядных итерационных алгоритмов МГУА со вложенными структурами. Оригинальный алгоритм предложен Олегом Шелудько в [9], а его модификация – МАКСО, – Александром Павловым в [10]. В работе описывается как оригинальный алгоритм, так и его модификация.

Модифицированный упрощённый алгоритм МГУА. Для формализации описания алгоритма введём следующие обозначения:

$\{x_1, \ldots x_m\}$ – множество входов (m векторных входных аргументов) размерности n, где n – количество точек наблюдений;

y – выходной вектор, размерности n;

W – исходная выборка данных размерности $n \times m$, $|W| = n$;

A – обучающая выборка, $|A| = N_A$;

B – проверочная выборка, $|B| = N_B$.

В данном алгоритме можно выделить два этапа: этап формирования обобщённых переменных и этап итераций.

Этап формирования обобщённых переменных. Для формирования обобщённых переменных задаётся множество функциональных преобразований FT (Functional Transformations). В алгоритме это множество содержит лишь одну функцию: взятие обратной величины от аргумента, т.е. $1/x_i$, $i = 1, \ldots, m$. Таким образом, после первого этапа у нас есть возможность расширить исходное множество переменных в 2 раза. Дальнейшее описание алгоритма приведём при расширенном множестве исходных аргументов.

Этап итераций. Для описания данного этапа удобно выделить следующие его шаги: формирование частного описания первого ряда итерационного процесса; формирование частных описаний последующих рядов.

При формировании частного описания на каждой итерации производится центрирование x_i, $i = 1, \ldots, 2m$ и y на обучающей выборке A, с целью расчёта только двух коэффициентов:

$$\tilde{x}_{ji} = x_{ji} - \bar{x}_i,\ \tilde{y}_j = y_j - \bar{y},\ j = \overline{1, N_A}$$

$$\bar{x}_i = \frac{1}{N_A}\sum_{j=1}^{N_A} x_{ji}\ ,\ \bar{y} = \frac{1}{N_A}\sum_{j=1}^{N_A} y_j$$

Формирование частного описания первого ряда. На первом ряде строится модель вида:

$$\tilde{y} = \theta_1 \tilde{z}_1,$$

где процесс формирования аргумента $\tilde{z}_1$ состоит из следующих шагов:

Шаг 1. Строятся $2m$ моделей вида:

$$\tilde{y} = \theta_1 \tilde{x}_i,\ i = \overline{1,2m}$$

Верхний индекс у переменных $\hat{\tilde{y}}^1$ и $\hat{\theta}_1^1$ – текущий номер ряда.

Шаг 2. Из этих моделей выбирается F лучших по критерию селекции алгоритма. Число F является так называемой свободой выбора данного алгоритма, и задаётся заранее. Число F используется только для процедуры формирования аргумента $\tilde{z}_r$ текущего ряда r. Общее количество итераций (рядов) R задаётся заранее. На рис.1 показан процесс проектирования вектора $\tilde{y}$ на вектор $\tilde{x}_i$.

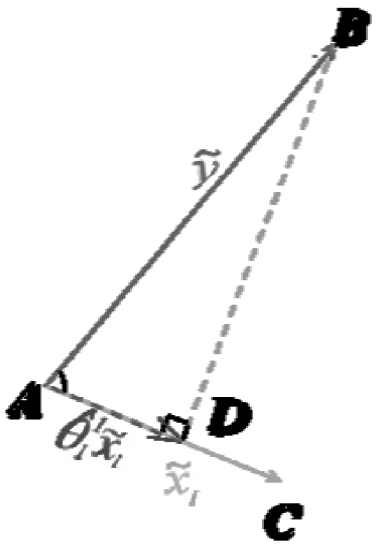

Рис.1

Оценку $\hat{\theta}_1$ находим по формуле:

$$\hat{\theta}_1^1 = \sum_{j=1}^{N_A} \tilde{x}_{ji}\tilde{y}_j \Big/ \sum_{j=1}^{N_A} \tilde{x}_{ji}\tilde{x}_{ji}$$

Приближённое решение будет иметь вид:

$$\hat{\tilde{y}}^1 = \hat{\theta}_1^1 \tilde{x}_i$$

Критерий селекции алгоритма имеет следующий вид:

$$Cr = \beta(\alpha \cdot Err_A + (1-\alpha)Err_B) + (1-\beta)\cdot\left|\frac{Err_A - Err_B}{Err_A + Err_B}\right|, \quad (1)$$

$$Err_A = \sum_{j=1}^{N_A}(\tilde{y}_j - \hat{\tilde{y}}_j^1)^2 \Big/ \sum_{j=1}^{N_A}\tilde{y}_j^2, \quad Err_B = \sum_{j=1}^{N_B}(\tilde{y}_j - \hat{\tilde{y}}_j^1)^2 \Big/ \sum_{j=1}^{N_B}\tilde{y}_j^2, \quad (2)$$

где: α – вес ошибки обучения; $(1-\beta)$ – вес учёта рассогласования ошибок обучения и проверки; Err_A – ошибка обучения; Err_B – ошибка проверки.

Коэффициенты α, β, а также длины выборок N_A и N_B задаются заранее. Следует отметить, что использование коэффициента $(1-\beta)$ имеет смысл только при равной длине выборок A и B.

Шаг 3. Выбрав F лучших по критерию селекции моделей, для каждой из них осуществляется процедура доопределения выбранного аргумента x_i, $i = 1, \ldots, F$ до окончательного аргумента $\tilde{z}_i$ комбинаторным способом. Для описания процедуры доопределения введём следующие обозначения:

k – максимальное количество произведений, которое допускается реализовать при формировании аргумента текущего ряда $\tilde{z}_r$, $r = 1, \ldots, R$. Параметр k задаётся заранее.

arg_i – текущий аргумент реализованный посредством домножения, $i = 1, \ldots, F$

Алгоритм доопределения следующий:

1. $i = 0$
2. Выбрать текущий x_i
3. Если $i > F$, то выбрать лучший по значению критерия $\tilde{z}_1 = \tilde{z}_i = \tilde{\arg}_i$, в качестве аргумента первого ряда, иначе перейти на шаг 4;
4. $l = 0$;
5. Если $l > k$, то i++ и перейти на 2, иначе перейти на шаг 6;
6. $q = 0$;
7. Если $q > 2m$, перейти на шаг 11, иначе перейти на шаг 8;
8. Умножить текущий аргумент arg_i (при $l = 0$, $arg_i = x_i$) на x_q. Отнормировать полученный аргумент $arg_{iq} = arg_i \cdot x_q$ по выборке A – $\tilde{\arg}_{iq}$;
9. Построить модель вида:

$$\tilde{\hat{y}}^1_{iq} = \hat{\theta}^1_1 \tilde{\arg}_{iq}$$

10. Рассчитать значение критерия селекции для $\tilde{\hat{y}}^1_{iq}$ модели и сохранить её, если это значение меньше всех значений критериев уже построенных моделей $\tilde{\hat{y}}^1_{ij}$, $j = 1, \ldots, q$; q++, перейти на шаг 7.
11. Если в цикле 7-10 нашлась модель, улучшающая значение критерия, то l++ и перейти на шаг 5, иначе i++ и перейти на шаг 2.

Таким образом, в данном алгоритме на следующий ряд передаётся только одна модель, т.е. свобода выбора, равна 1.

Оценку свободного члена для $\hat{y}^1$ находим по формуле:

$$\hat{\theta}_0^1 = \overline{y} - \hat{\theta}_1^1 \overline{z}_1 = \overline{y} - \varphi_1$$

Формирование частных описаний последующих рядов. На последующих рядах строится модель вида:

$$\tilde{y}^{r+1} = \theta_1 \hat{\tilde{y}}^r + \theta_2 \tilde{z}_{r+1}$$

где процесс формирования аргумента $\tilde{z}_{r+1}$ состоит из следующих шагов:

Шаг 1. Строятся 2*m* моделей вида:

$$\tilde{y}_i^{r+1} = \theta_1 \hat{\tilde{y}}^r + \theta_2 \tilde{x}_i, i = \overline{1,2m} \qquad (3)$$

Шаг 2. Из этих моделей снова выбирается *F* лучших по критерию селекции алгоритма. На рис.2 показан процесс определения решения *r*+1 ряда.

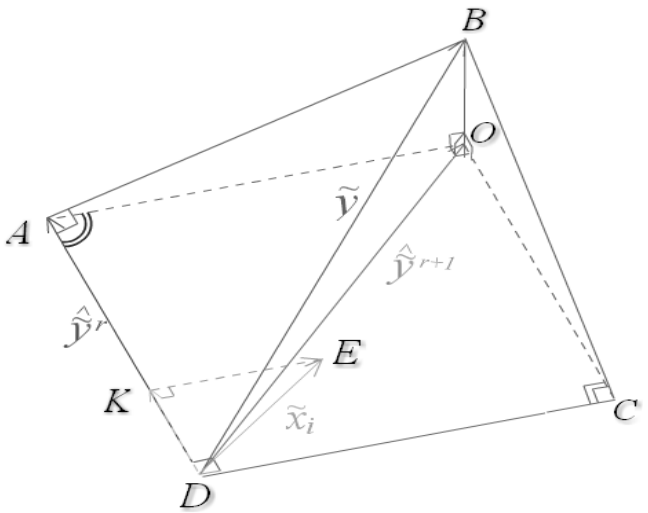

Рис. 2

Расчёт оценок параметров $\hat{\theta}_1^{r+1}, \hat{\theta}_2^{r+1}$ и решения следующего (*r*+1) реализуется следующим образом:

1. Находится вектор $\overrightarrow{KE}$, ортогональный к вектору предыдущего решения $\hat{\tilde{y}}^r$, для этого:

Вычисляется проекция вектора $\overrightarrow{DE} = \tilde{x}_i$ на вектор $\overrightarrow{DA} = \hat{\tilde{y}}^r$:

$$\tilde{x}_i = b_{r+1} \hat{\tilde{y}}^r,\ \hat{b}_{r+1} = \sum_{j=1}^{N_A} \hat{\tilde{y}}_j^r \tilde{x}_{ji} \Big/ \sum_{j=1}^{N_A} \hat{\tilde{y}}_j^r \hat{\tilde{y}}_j^r,$$

Находится вектор $\overrightarrow{KE}$ как:

$$\overrightarrow{KE} = \overrightarrow{DE} - \overrightarrow{DK} = \tilde{x}_i - \hat{b}_{r+1} \hat{\tilde{y}}^r$$

2. Вычисляется проекция вектора $\tilde{y}$ на направление $\overrightarrow{KE}$:

$$\tilde{y} = c_{r+1}\overrightarrow{KE},$$

$$\hat{c}_{r+1} = \frac{\sum_{j=1}^{N_A} \tilde{y}_j \overrightarrow{KE}_j}{\sum_{j=1}^{N_A} \overrightarrow{KE}_j \overrightarrow{KE}_j}$$

3. Находится решение $\tilde{y}^{r+1}$ как:

$$\tilde{y}^{r+1} = \overrightarrow{DA} + \overrightarrow{DC} = (1 - \hat{c}_{r+1}\hat{b}_{r+1})\hat{\tilde{y}}^r + \hat{c}_{r+1}\tilde{x}_i, \quad (4)$$

где $\hat{\theta}_1^{r+1} = (1 - \hat{c}_{r+1}\hat{b}_{r+1})$, $\hat{\theta}_2^{r+1} = \hat{c}_{r+1}$

Таким образом, оценки коэффициентов имеют вид:

$$\hat{\theta}_l^{r+1} = (1 - \hat{c}_{r+1}\hat{b}_{r+1})\hat{\theta}_l^r, \; l = \overline{1, r}$$

Шаг 3. Для каждой из выбранных *F* моделей вида (3) осуществляется процедура доопределения выбранного аргумента x_i, i = 1, …, *F* до окончательного аргумента до $\tilde{z}_{r+1}$.

Оценка свободного члена также находиться по рекуррентной формуле:

$$\hat{\theta}_0^{r+1} = \bar{y} - \hat{c}_{r+1}\bar{z}_{r+1} - (1 - \hat{c}_{r+1}\hat{b}_{r+1})\varphi_r, \; \varphi_r = \hat{c}_r\bar{z}_r + (1 - \hat{c}_r\hat{b}_r)\varphi_{r-1},$$

где $\varphi_1 = \hat{c}_1\bar{z}_1 = \hat{\theta}_1^1\bar{z}_1$

Если вектор $\tilde{z}_{r+1}$ оказался ортогональным вектору $\tilde{y}^r$, то коэффициент $\hat{b}_{r+1}$= 0 в (3), и формула (4) примет вид:

$$\tilde{y}^{r+1} = \hat{\tilde{y}}^r + \hat{\theta}_2^{r+1}\tilde{x}_i$$

При этом коэффициенты предыдущих рядов не корректируются, а свободный член вычисляется по формуле:

$$\hat{\theta}_0^{r+1} = \bar{y} - \hat{c}_{r+1}\bar{z}_{r+1} - \varphi_r$$

Если вектор $\tilde{z}_{r+1}$ колинеарен $\tilde{y}^r$, то формируется следующий вектор $\tilde{z}_{r+1}$.

Если же вектор $\tilde{z}_{r+1}$ лежит в плоскости $\tilde{y}^r$ и $\tilde{z}_r$, то алгоритм оценивания параметров найдёт $\tilde{y}^{r+1} = \tilde{y}^r$, т.е. данный алгоритм не проверяет возможность линейной зависимости векторов $\tilde{z}_{r+1}, \tilde{y}^r$ и $\tilde{z}_r$.

Сходимость алгоритма оценивания параметров. Для доказательства сходимости алгоритма оценивания параметров к оценке по МНК, рассмотрим

рис. 1. Пусть на рис. 1 $r = 1$. Очевидно, что для первых двух итераций, векторы $\widehat{\tilde{y}}^2_{МАКСО}, \widehat{\tilde{y}}^2_{МНК}$ полученные с помощью описанного алгоритма, и алгоритма МНК будут совпадать. Однако, если продолжить изложенную процедуру построения, вектор $\widehat{\tilde{y}}^3_{МАКСО}$, построенный на 3-м ряду, уже не совпадёт (в общем случае) с вектором $\widehat{\tilde{y}}^3_{МНК}$ ($\widehat{\tilde{y}}^3_{МНК} = \tilde{y}$ при условии, что вектор $\tilde{y}$ расположен в трёхмерном пространстве), полученным по МНК. Это происходит из-за того, что вектор $\widehat{\tilde{y}}^3_{МАКСО}$ ищется как проекция в плоскость векторов $\widehat{\tilde{y}}^2$ и $\tilde{z}_3$, а не в гиперплоскость векторов $\tilde{z}_1, \tilde{z}_2, \tilde{z}_3$. Несмотря на это, при бесконечном количестве рядов данного алгоритма, вектор $\widehat{\tilde{y}}^r_{МАКСО}$ будет сходиться к вектору $\tilde{y}$. Докажем это. Рассмотрим рис.1. Очевидно, что невязка r-го (1-го) это вектор $\overrightarrow{AB} = \tilde{y} - \widehat{\tilde{y}}^r$. Невязка же следующего, (r + 1)-го (2-го) ряда – это вектор $\overrightarrow{OB} = \tilde{y} - \widehat{\tilde{y}}^{r+1}$. Теперь рассмотрим треугольник ΔAOB, в нём OB – это катет, а AB – гипотенуза. Таким образом, с каждой итерацией расстояние (невязка) между векторами $\tilde{y}^r$ и $\tilde{y}$ сокращается. Откуда следует, что при бесконечном количестве итераций эти векторы совпадут.

Останов алгоритма. При использовании классического критерия регулярности ($\alpha = 0$, $\beta = 1$ в формуле (1)) останов алгоритма происходит, тогда когда значение $Cr_{r+1} > Cr_r$. При использовании большого коэффициента α, останов алгоритма определяется числом рядов: оператор задаёт достаточно большое количество рядов, и смотрит, насколько уменьшается ошибка моделирования от ряда к ряду. Если эта разница становиться меньше наперёд заданного числа ε, оператор выбирает модель соответствующую этому количеству рядов.

Особенности МУА МГУА

1. В алгоритме реализован режим перемешивания точек при оценивании параметров следующего ряда. В данном режиме построения модели нет точно определённых точек, которые принадлежат выборкам A и B. При использовании данного режима можно варьировать параметром повторений R_p, определяющим количество расчётов оценок параметров $\widehat{\theta}_1^{r+1}, \widehat{\theta}_2^{r+1}$ при разном наборе случайно перемешанных точек.
2. На следующий ряд передаётся только одна модель, т.е. свобода выбора, в классическом смысле для этого алгоритма равна 1.

   Особенности генератора структур
3. Усечённая целесообразным образом процедура формирования аргумента $\tilde{z}_{r+1}$ (хотя и с возможными повторами) заключающаяся в последовательном наращивании мультипликативным образом сложности аргумента до тех пор, пока значение критерия селекции Cr уменьшается.

   Особенности алгоритм оценивания параметров

4. Идея рекуррентной оценки параметров, основанная на проектировании вектора $\tilde{y}$ не в гиперплоскость всех аргументов, а плоскость $\tilde{y}^r$ и $\tilde{z}_{r+1}$.
5. Оценивание параметров использует центрирование данных с целью расчёта только двух параметров $\hat{\theta}_1^{r+1}, \hat{\theta}_2^{r+1}$.

   Класс моделей
6. Алгоритм реализует класс полиномиальных частных описаний указанного максимального порядка. В алгоритме реализовано преобразование исходных аргументов $FT = \{1/x\}$, с помощью которого в синтезируемую модель можно ввести обратные величины переменных.
7. Имеется возможность построения полиномиальных моделей множественной регрессии.

   Критерий Селекции
8. Предложен комбинированный критерий селекции, позволяющий, как частный случай, задавать в качестве внешнего дополнения критерий регулярности.

Модифицированный алгоритм с комбинаторной селекцией и ортогонализацией переменных. Отличительными особенностями МАКСО от МУА МГУА являются:

1. расширение множества функциональных преобразований: добавлены два вида функциональных преобразований исходных переменных:

$$FT = \{\sqrt[3]{x}, \frac{1}{\sqrt[3]{x}}\};$$

2. увеличена гибкость моделирования: добавлена возможность указания максимального количества мультипликативных членов отдельно для каждого ряда, т.е. можно задать k_r, $r = 1,\ldots, R$;
3. реализован удобный программный интерфейс, позволяющий легко назначить выход и входы перед началом моделирования;
4. добавлены два новых критерия селекции: «Гладкости » и «Дробление».

Для описания этих критериев введём обозначения:

Model – это массив выхода текущей модели длинной $2n - 1$, имеющий следующую структуру: значения в нечётных индексах массива содержат значение модели в соответствующей точке i, $i = 1,\ldots, n$; значения в чётных индексах массива содержат значение модели в точке, находящейся посередине между табличными точками, индексы которых являются соседними (нечётными) к этой точке.

I – множество чётных индексов массива *Model*.

Критерий «Гладкости»

1. Вычисляется ошибка гладкости на всех точках:

$$Err_{Smoothing} = \frac{\sum_{i \in I} (Model[i] - (Model[i-1] + Model[i+1])/2)^2}{\sum_{i \in I} ((Model[i-1] + Model[i+1])/2)^2}$$

2. Рассчитываются ошибки обучения Err_A и проверки Err_B по формулам (2).

3. Вычисляется критерий селекции:

$$Cr = \beta(\alpha \cdot Err_A + (1-\alpha)Err_B) + oscil \cdot Err_{Smoothing} + (1-\beta) \cdot \left| \frac{Err_A - Err_B}{Err_A + Err_B} \right|,$$

где параметр *oscil* – это вес осцилляции, задаётся заранее.

Критерий «Дробления»

Пусть Y – множество индексов выходного массива (вектора) $\tilde{y}$.

1. Вычисляется ошибка дробления на всех точках:

$$Err_{Splitting} = \frac{\sum_{i \in I, j \in Y} (Model[i] - (\tilde{y}[j] + \tilde{y}[j+1])/2)^2}{\sum_{j \in Y} ((\tilde{y}[j] + \tilde{y}[j+1])/2)^2}$$

2. Рассчитываются ошибки обучения Err_A и проверки Err_B по формулам (2).

3. Вычисляется критерий селекции:

$$Cr = \beta(\alpha \cdot Err_A + (1-\alpha)Err_B) + splitting \cdot Err_{splitting} + (1-\beta) \cdot \left| \frac{Err_A - Err_B}{Err_A + Err_B} \right|,$$

где параметр *splitting* – это вес дробления, задаётся заранее.

Выводы.

Основными преимуществами данного алгоритма являются:

- Идея построения модели, методом вложенных структур, оценивая при этом на каждом ряду лишь два коэффициента по достаточно простым формулам;
- Усечённая целесообразным образом процедура формирования следующего аргумента модели, позволяющая взять в рассмотрение значительно более сложные структуры, нежели те, что перебираются в алгоритмах COMBI и MULTI [12].

- Возможность ввода в рассмотрение обратного степени и кубического корня от исходного аргумента.

Однако данные алгоритм и не лишён недостатков:

- из формул оценки коэффициентов данного алгоритма видно, что они зависят от количества точек выборок *A* и *B*. Это говорит о том, что при больших исходных выборках (N > 3000), алгоритм будет существенно увеличивать время построения модели;
- алгоритм не реализует принцип неокончательных решений, поскольку, фактическая свобода выбора в нём равна 1.

В дальнейшем планируется работа в направлении ликвидации двух выявленных недостатка.

Список литературы

[1] Ivakhnenko, A.G.: Polynomial Theory of Complex System. IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-1, No. 4, Oct. 1971, pp. 364-378.

[2] Petr Buryan. Enhanced MIA-GMDH Algorithm. Proceedings of IWIM 2007, p. 144-155.

[3] Marcel Jirina, Marcel Jirina. Genetic Selection and Cloning in GMDH MIA Method. Proceedings of IWIM 2007, pp.165-171.

[4] Godfrey C. Onwubolu. Design of Hybrid Differential Evolution and Group Method of Data Handling for Inductive Modeling. Proceedings of IWIM 2007, pp. 87-95.

[5] Oleg Kovarik, Pavel Kordık. Optimizing Models Using Continuous Ant Algorithms. Proceedings of ICIM 2008, pp. 124-128.

[6] Godfrey Onwubolu, Anurag Sharma, Ashwin Dayal, Deepak Bhartu, Amal Shankar, Kenneth Katafono. Hybrid Particle Swarm Optimization and Group Method of Data Handling for Inductive Modeling. Proceedings of ICIM 2008, pp. 96-103.

[7] Pavel Kordik. Regularization of Evolving Polynomial Models. Proceedings of IWIM 2007, pp. 294-301.

[8] Tadashi Kondo, Junji Ueno. Multi-Layered GMDH-Type Neural Network Self-Selecting Optimum Neural Network Architecture and Its Application to Nonlinear System Identification. Proceedings of IWIM 2007, pp. 55-62.

[9] Oleg Sheludko. Self-organization of mathematical models for solving particular control and reliability problems. Dissertation for seeker of Ph.D. degree. Kiev. 1975, 166 p.

[10] Vanin V. V., Pavlov Alex. V.: Development and application of self-organization algorithms for modeling of complex processes and objects which are represented by the point former, Proceedings of Tavria State agrotechnical academy. Pub. 4, Vol. 24, Melitopol, 2004, 51-56. (In Ukrainian).

[12] А.Г. Ивахненко, В.С. Степашко. Помехоустойчивость моделирования. Изд. «Наукова думка», Киев 1985.