

О.Г. Мороз

**Переборный алгоритм МГУА с генетическим поиском оптимальной модели**

Представлена архитектура обобщенного итерационного алгоритма метода группового учета аргументов, частными случаями которого есть алгоритмы многорядного и релаксационного типов, а также разновидности итерационно-комбинаторных алгоритмов. На основе предложенной методики численного анализа эффективности итерационных алгоритмов выполнено их сравнительное исследование и вопрос сходимости к истинной модели.

**Ключевые слова:** комбинаторный алгоритм МГУА (КОМБИ), генетический алгоритм, гибридный алгоритм МГУА.

Представлено архітектуру узагальненого ітераційного алгоритму методу групового урахування аргументів, окремими випадками якого є алгоритми багаторядного та релаксаційного типів, а також різновиди ітераційно-комбінаторних алгоритмів. На основі запропонованої методики чисельного аналізу ефективності ітераційних алгоритмів виконано їх порівняльне дослідження і питання збіжності до істинної моделі.

**Ключові слова:** комбінаторний алгоритм МГУА (КОМБИ), генетичний алгоритм, гібридний алгоритм МГУА.

**Введение.** Комбинаторный алгоритм МГУА или КОМБИ впервые был разработан и описан в [1, 2]. В этом алгоритме выполняется полный перебор всех частных моделей из заданного базиса и выбирается лучшая из них по заданному критерию селекции (или по ансамблю критериев), исключая возможность потери оптимальной модели и делает его надежным и простым. Алгоритм – эффективное средство решения задач структурно-параметрической идентификации, прогнозирования, построения моделей объектов и процессов по экспериментальным данным в условиях неопределенности и неполноты информации. В частности, КОМБИ успешно применяется при моделировании и прогнозировании в экономике [3, 4], социологии [5, 6], медицине [7], естественных науках [8] и т.д. Задачи, решаемые с помощью КОМБИ, по своей сути относятся к классу задач дискретной оптимизации.

Основной недостаток комбинаторного алгоритма – экспоненциальный рост времени решения задачи с увеличением количества входных переменных (каждая дополнительная переменная увеличивает время расчета вдвое). В частности, при современных технических возможностях он успешно применяется лишь для решения задач с менее чем 30-ю входными переменными (аргументами), что требует значительных расходов времени и вычислительных ресурсов.

Этот недостаток в некоторой степени устраняют переборные алгоритмы с детерминированным [9–11] и случайным [12] неполным направленным перебором. Детерминированные алгоритмы способны существенно уменьшить время

и вычислительные ресурсы, необходимые для поиска оптимальной модели, в сравнении с обычным комбинаторным алгоритмом. Однако такие алгоритмы имеют достаточно сложные схемы поиска оптимальной модели и требуют значительных усилий для их практической реализации. В отличие от этого алгоритм со случайным перебором совмещает КОМБИ с генетическим алгоритмом (ГА), который имеет прозрачную структуру и простую реализацию.

В статье обобщаются и развиваются последние результаты в разработке и оценивании эффективности гибридного переборного алгоритма КОМБИ-ГА, в котором для перебора моделей и ускоренного поиска оптимальной из них используются генетические операторы. Продемонстрировано влияние генетических операторов на эффективность работы алгоритма КОМБИ-ГА на тестовых и реальной задаче.

**Общая структура и некоторые ограничения комбинаторного алгоритма**

В структуре комбинаторного алгоритма можно выделить три основных блока [13]:

- преобразование исходных данных в соответствии с избранной системой опорных (базисных) функций, в которой подыскивается модель (базис может быть полиномиальным, тригонометрическим, в виде разностных уравнений и т.д.);
- генерирование и перебор моделей разной структуры в избранном базисе;
- вычисление значений некоторого критерия селекции, имеющего свойства внешнего дополнения, и последовательный отбор частных моделей, лучших по этому критерию.

В различных переборных алгоритмах МГУА используются фактически одни и те же критерии селекции и системы опорных функций. Существенное отличие этих алгоритмов между собой – структура «генератора моделей», т.е. схема вычислений, реализуемая во втором блоке. Рассмотрим особенности первых двух блоков в случае полиномиальных опорных функций.

**Блок формирования базиса.** Если задано  $r$  входных переменных  $z_1, z_2, \dots, z_r$  моделируемого объекта и максимальная степень полного полинома равняется  $p$ , то число слагаемых  $m$  в этом полиноме можно найти по формуле:

$$m = m_{r,p} = \frac{(r+p)!}{r!p!}.$$

Например,  $m_{6,2} = 28$ ,  $m_{6,4} = 210$ ,  $m_{6,6} = 924$ ,  $m_{6,8} = 3003$ ,  $m_{8,2} = 45$ ,  $m_{10,2} = 66$ ,  $m_{12,2} = 91$ ,  $m_{14,2} = 120$ . Эти цифры дают определенное представление о скорости роста  $m$  в зависимости от  $r$  и  $p$ .

Если  $p_i = (p_{i1}, p_{i2}, \dots, p_{ir})$  – вектор степеней всех  $r$  переменных  $i$ -го члена полного полинома, где  $p_{ij} \in \{0, 1, \dots, p\}$  и  $\sum_{j=1}^r p_{ij} \leq p$ , то такой полином можно записать в виде:

$$y = a_1x_1 + a_2x_2 + \dots + a_mx_m,$$

$$x_i = x_i(z_1, \dots, z_r) = \prod_{j=1}^r z_j^{p_{ij}} \quad (i = 1, 2, \dots, m),$$

где все обобщенные линейные аргументы  $x_i$  образуют базисный набор функций для комбинаторного алгоритма.

**Блок генерирования частных моделей.** Основные операции, осуществляемые в этом блоке, следующие: формирование структуры очередной частной модели и соответствующей системы нормальных уравнений, а также ее решение (оценка коэффициентов модели).

Структуры частных моделей формализуются с помощью двоичного структурного вектора  $d = (d_1, d_2, \dots, d_m)$ : если элемент  $d_i$  принимает значение единицы, то соответствующий  $i$ -й аргумент включается в модель, если ноль – не включается ( $i = 1, \dots, m$ ). Тогда общее количество частных моделей равняется  $R_m = 2^m - 1$ . Так, например,  $R_{35} = 34359738367$ , т.е. при полном переборе для нахождения наилучшей модели необходимо для каждой из свыше 34 миллиардов частных моде-

лей сформировать и решить соответствующую систему нормальных уравнений. Такой объем вычислений нуждается в очень больших вычислительных и временных ресурсах.

Изменение состояний вектора  $d$  можно организовать многими способами, самые известные из которых описаны в [14]. Их общей характеристикой есть то, что они используют определенные детерминированные правила, по которым выполняется перебор. В данной статье используется эвристический алгоритм генерации состояний вектора  $d$  со случайным перебором на основе генетических операторов, впервые описанный и исследован в [12].

### Генетические алгоритмы и операторы генерации потомков

Генетический алгоритм – это один из метаэвристических алгоритмов глобальной оптимизации, который был сконструирован в результате обобщения и имитации в искусственных системах таких свойств живой природы, как естественный отбор, приспособляемость к изменяемым условиям среды, наследование потомками жизненно необходимых свойств от родителей и т.д. Основные принципы ГА сформулированы Джоном Холландом [15] и хорошо описаны во многих работах. Формально ГА можно представить таким образом:

$$ГА = \langle P_0, M, L, F, G, s \rangle,$$

где  $P_0 = (a_1^0, \dots, a_M^0)$  – начальная популяция;  $a_i^0$  – претендент на решение задачи оптимизации, представленный в виде хромосомы;  $M$  – целое число (размер популяции);  $L$  – целое число (длина каждой хромосомы популяции);  $F$  – целевая функция или функция пригодности (ФП);  $G$  – множество генетических операторов;  $s$  – правило завершения алгоритма.

Хромосомы состоят из генов, которые обычно кодируются значениями из двоичного алфавита  $\{0, 1\}$ , однако можно использовать и другие алфавиты – буквенные, десятичные и др. В статье рассматривается лишь двоичное кодирование хромосом.

Входными данными для любого ГА есть начальная популяция – конечное множество хромосом (элементов, особей, экземпляров и т.п.)

$P_0$ , каждая из которых представляет возможный вариант решения задачи. Далее из родительских хромосом в  $P_0$  с помощью определенных генетических правил образуется первая популяция потомков  $P_1$ , аналогично из популяции  $P_1$  образуется следующая популяция  $P_2$  и так далее. Процесс продолжается до тех пор, пока не будет выполнено заданное правило завершения алгоритма. Важной особенностью работы ГА есть то, что с каждым шагом среднее значение ФП текущей популяции уменьшается и приближается к решению оптимизационной задачи.

Центральную роль в работе ГА играют *операторы создания новых особей (потомков)* из имеющихся в текущей популяции, которые можно разделить на две основные группы. Операторам первой группы для создания двух потомков необходимы две особи (родители) и правила обмена генетическим материалом между ними. Операторы второй группы формируют одного потомка из одной особи, модифицируя ее по определенным правилам. Примерами операторов первой группы есть кроссинговер, транслокация, сегрегация и др., ко второй группе принадлежат операторы мутации, инверсии, транспозиции и др. [16]. Важнейшее из них – операторы кроссинговера (ОК) и мутации (ОМ), выполняемые не во всех случаях формирования потомков, а с заданной вероятностью, – близкой к единице для операторов кроссинговера и к нулю для большинства операторов мутации. В целом жестких ограничений в использовании разных моделей образования потомков фактически нет, потому что при решении задач нет необходимости полностью копировать законы природы и ограничиваться только ими. Целесообразнее пользоваться здравым смыслом и максимально учитывать специфику задачи.

Рассмотрим краткое описание операторов, исследуемых в статье.

*Одноточечный оператор* кроссинговера (оОК) [16]. Случайно из популяции выбирают две хромосомы, дальше произвольным образом генерируется так называемая точка кроссинговера  $l_k \in [1, L-1]$ , где  $L$  – длина хромосом в популяции. В результате кроссинговера (скрещивания) пары родительских хромосом относительно

$l_k$  получим такую пару потомков: первый потомок – хромосома, которая на позициях от единицы до  $l_k$  содержит гены первого из родителей, а на позициях от  $l_k+1$  до  $L$  – гены второго родителя; второй потомок – это хромосома, которая, соответственно, на позициях от единицы до  $l_k$  содержит гены второго из родителей, а на позициях от  $l_k+1$  до  $L$  – гены первого из родителей.

*Равномерный оператор кроссинговера* (рОК) [16]. Одно из существующих описаний этого ОК следующее. Случайно образуют маску скрещивания – строку из нулей и единиц, длина которой равняется длине хромосомы. Маска указывает, какие гены должны быть унаследованы от первого из родителей, а какие – от второго, при этом два новых потомка образуются по такому правилу: единица в  $i$ -й позиции маски означает, что  $i$ -й элемент первого родителя следует расположить на  $i$ -м месте первого потомка, а ноль – что  $i$ -й элемент второго родителя следует расположить на  $i$ -м месте первого потомка. Считая первого родителя вторым, а второго – первым, по описанному правилу получают второго потомка. Маска скрещивания может быть одна для всех или своя для каждой пары родителей.

*Кольцевой оператор кроссинговера* (кОК) [17]. В этом кроссинговере родительские хромосомы объединяются началами и концами, формируя кольцо, дальше генерируется случайная точка его разрыва. Прямая, проходящая через эту точку и центр кольца, разделяет его на две части. Первый потомок формируется по часовой стрелке от точки кроссинговера, а второй – против нее.

*Универсальный оператор кроссинговера* (уОК) [16]. Аналогично равномерному кроссинговеру вводится двоичная маска, длина которой равняется длине заданных хромосом. Первый потомок образуется побитовым сложением элементов хромосомы первого родителя и маски по таким правилам:  $0+0=0$ ,  $0+1=1$ ,  $1+1=0$ . Для получения второго потомка действуют аналогичным способом со вторым родителем.

*Многоточечный оператор мутации* (мТОМ) [18]. В хромосоме случайным образом выбирают несколько генов (позиций структурного вектора), причем их количество может быть случайным, и значения всех выбранных генов изменяются на противоположные.

## Гибридный алгоритм КОМБИ-ГА

Алгоритм КОМБИ-ГА поэтапно формирует множество наиболее перспективных структур частных моделей и находит оптимальную из них с использованием приведенных выше генетических операторов. Формально этот алгоритм можно описать так:

КОМБИ-ГА =  $\{Z, y, f, D, CR, M, H, G, F, k\}$ ,

где  $Z[n \times r]$  – матрица измерений вектора входных переменных моделируемого объекта, здесь  $r$  – количество входных переменных,  $n$  – число точек измерения;  $y[n \times 1]$  – вектор измерений выходной переменной моделируемого объекта;  $f[m \times 1]$  – вектор базисных (опорных) функций от входных переменных, где  $m$  – количество базисных функций;  $D$  – заданное правило разделения матрицы измерений  $X[n \times m]$  опорного набора аргументов (элементов вектора базисных функций) и вектора  $y[n \times 1]$  на обучающую и проверочную части;

$CR$  – внешний критерий селекции, основанный на указанном разделении выборки  $(X, y)$ ;

$M$  – размер популяции двоичных хромосом, т.е. закодированных структур частных моделей;

$H$  – размер текущей популяции из лучших моделей,  $H < M$ ;

$G$  – множество генетических операторов;

$k$  – критерий остановки ГА;

$F$  – количество наилучших частных моделей (свобода выбора), которые отслеживаются на всех этапах работы алгоритма,  $1 \leq F \leq H$ .

Алгоритм реализуется с помощью следующих шагов:

Шаг 1. Используя матрицу измерений вектора входных переменных  $Z$  и вектор базисных функций  $f$ , вычисляем матрицу измерений  $X[n \times m]$  опорного набора аргументов и разделяем ее согласно правилу  $D$  на обучающую  $X_A[n_A \times m]$  и проверочную  $X_B[n_B \times m]$  подматрицы ( $n_A + n_B = n$ ); точно так же разделяем и выходной вектор измерений  $y[n \times 1]$ . Очевидно, что в случае линейного полинома ( $p = 1$ ) матрицы  $X$  и  $Z$  тождественны (при этом  $m = r$ ).

Шаг 2. Генерируем случайным образом  $M$  двоичных хромосом, длина каждой из которых равняется  $m$ . Получаем начальную популяцию  $P_0$  генетического алгоритма.

Шаг 3. Используя обучающую матрицу базисных (эквивалентных) аргументов  $X_A$  и выходной вектор  $y_A$ , вычисляем по МНК или другим методом коэффициенты каждой из частных моделей популяции.

Шаг 4. Используя проверочную матрицу аргументов  $X_B$  и выходной вектор  $y_B$ , вычисляем для каждой частной модели значение внешнего критерия  $CR$ , который есть целевой функцией ГА.

Шаг 5. Образует текущую популяцию размера  $H$  ( $H < M$ ) частных моделей – хромосом с лучшими значениями критерия  $CR$  для дальнейшего формирования потомков. Кроме того, отбираем из нее  $F$  лучших частных моделей, являющихся потенциальным решением задачи построения (самоорганизации) моделей.

Шаг 6. С помощью генетических операторов из текущей популяции образуем новую популяцию из  $M$  особей ( $M > H$ ).

Шаг 7. Проверяем заданный критерий остановки ГА. Если он выполняется, то переходим к шагу 8, иначе – к шагу 3.

Шаг 8. Из особей текущей популяции частных моделей размера  $H$  выбираем  $F$  наилучших.

Шаг 9. Конец.

В зависимости от особенностей решаемой задачи, определенные шаги этого алгоритма можно детализировать. Более того, могут появиться дополнительные шаги, например, в случаях с несколькими критериями селекции, или когда к обучающей и проверочной добавляются другие подвыборки.

## Вычислительные эксперименты и их результаты

Цель выполнения экспериментов – определение эффективности гибридного алгоритма КОМБИ-ГА в зависимости от применяемых способов формирования генетическими операторами потомков отобранных элитных особей.

При использовании в ГА какого-либо из четырех операторов кроссинговера (уОК, оОК, рОК, кОК), с ними в паре применялась трехточечная мутация (ЗТОМ), с которой алгоритм КОМБИ-ГА работает более эффективно [19].

Исследование проводилось для задач с тремя типами данных – тестовыми без шума, тес-

товыми с добавлением шума и реальными при таких общих условиях: поиск осуществляется в классе линейных моделей; функцией пригодности ГА есть критерий регулярности МГУА; размер начальной популяции – 100 особей; вероятность применения оператора кроссинговера 0,9, вероятность мутации, т.е. вероятность, с которой хромосомы подвергаются 3тОМ 0,2; критерием остановки ГА есть достижение абсолютного значения разницы между минимальными значениями ФП предыдущей и текущей популяций, которая равняется  $10^{-7}$ ; выборка входных данных разделялась на обучающую, проверочную и экзаменационную подвыборки в соотношении 0,5:0,3:0,2; техническая оснащенность – двухъядерный процессор (2,7 ГГц).

**Случай незашумленных данных.** Здесь рассматривались три задачи с количеством входных переменных 20, 50 и 200 аргументов. Примерно половина элементов полученного вектора входных переменных в каждом случае – истинные аргументы, т.е. именно они учитываются при вычислении значений выходной величины по наперед заданной формуле. Например, для набора данных с 20 входными аргументами модель, заданная для экспериментов, содержала восемь информативных аргументов и имела вид:

$$y(x) = 0,5x_2 + 0,2x_4 + 2,4x_5 - 0,6x_9 + 0,3x_{13} + 2,1x_{15} + 2x_{17} + 0,8x_{19}.$$

Матрица аргументов  $X$  и коэффициенты (параметры) модели были получены с помощью генератора равномерно распределенных случайных чисел, причем значения аргументов находились в пределах от нуля до 20, а значения параметров – в пределах от –1 до шести. Исследовалось влияние тех или иных генетических операторов на эффективность гибридного алгоритма, т.е. на полноту выявления истинной (точной) структуры модели и на быстродействие процесса построения модели.

Результаты исследований, приведенные в табл. 1, показывают, что эффективнее всего КОМБИ-ГА работает с оОК, при использовании которого наблюдалось довольно быстрое уменьшение минимального значения функции пригодности популяции в каждом поколении ГА, что свидетельствует о направленном характере поиска оптимальной модели.

**Таблица 1.** Эффективность работы КОМБИ-ГА в зависимости от использованного оператора ГА (данные без шума)

Количество входных переменных	Время нахождения оптимальной модели, с.				
	КОМБИ	КОМБИ-ГА			
		оОК	уОК	кОК	рОК
20	84,9	0,08	0,084	0,09	0,081
50	–	1,36	4,7	4,4	2,2
200	–	74,15	100,5	80,63	84,1

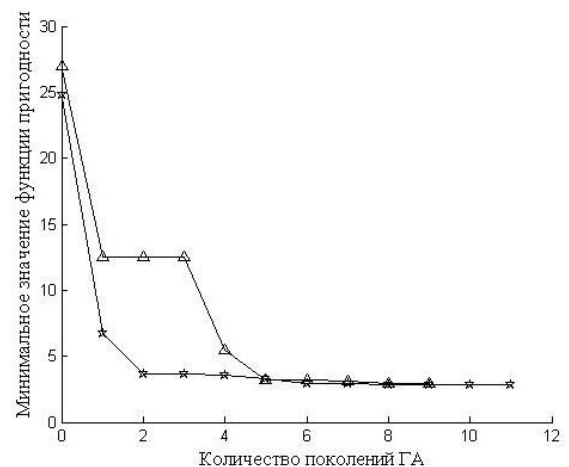
**Случай зашумленных данных.** Для случая 20 входных аргументов модель, заданная для экспериментов, была той же, что и в примере без шума.

При наличии 20 процентов шума в данных алгоритмы КОМБИ и КОМБИ-ГА с оОК нашли, соответственно, такие оптимальные модели с сохранением той же структуры, за исключением нескольких членов с лишними (неинформативными) аргументами, имеющими малозначимые коэффициенты:

$$y_1(x) = 0,5x_2 + 0,26x_4 + 2,39x_5 + 0,01x_7 - 0,61x_9 + 0,03x_{10} - 0,01x_{12} + 0,3x_{13} + 2,08x_{15} + 1,99x_{17} + 0,86x_{19};$$

$$y_2(x) = -0,01x_1 + 0,59x_2 + 0,16x_4 + 2,38x_5 - 0,02x_7 - 0,63x_9 + 0,04x_{12} + 0,39x_{13} - 0,05x_{14} + 2,14x_{15} + 2x_{17} + 0,74x_{19}.$$

При использовании других генетических операторов алгоритм КОМБИ-ГА работал менее эффективно. Графики сходимости КОМБИ-ГА в зависимости от генетических операторов представлены на рис. 1.



**Рис. 1.** Сходимость КОМБИ-ГА с использованием ГА с операторами (оОК+3тОМ) (☆) и (кОК+3тОМ) (Δ)

При наличии 50 процентов шума в данных КОМБИ нашел такую оптимальную модель:

$$y(x) = 0,49x_2 + 0,35x_4 + 2,37x_5 + 0,02x_7 - 0,62x_9 - 0,08x_{10} - 0,02x_{12} + 0,19x_{13} + 0,11x_{14} + 2,06x_{15} + 1,97x_{17} + 0,01x_{18} + 0,94x_{19}.$$

Используя оОК, алгоритм КОМБИ-ГА работал эффективнее всего и нашел следующую оптимальную модель:

$$y(x) = 0,48x_2 + 0,211x_4 + 2,312x_5 - 0,63x_9 + 0,23x_{13} + 2,14x_{15} + 1,48x_{17} + 0,867x_{19},$$

которая содержит все истинные аргументы без появления лишних.

Для набора данных с 50 входными аргументами (из которых 25 – истинные) эффективнее всего работал алгоритм КОМБИ-ГА с оОК, а оптимальная модель содержала 25 аргументов при 20 процентах и 24 при 50 процентах шума. Эффективность КОМБИ-ГА с разными генетическими операторами представлена в табл. 2. На рис. 2. показана сходимость КОМБИ-ГА с оОК.

Таблица 2. Эффективность работы КОМБИ-ГА в зависимости от операторов ГА

Количество входных переменных	Время нахождения оптимальной модели, с.				
	КОМБИ	20% шума		50% шума	
		оОК	кОК	оОК	кОК
20	85,2	0,158	0,17	0,185	0,2
50	–	0,51	0,7	0,53	0,8

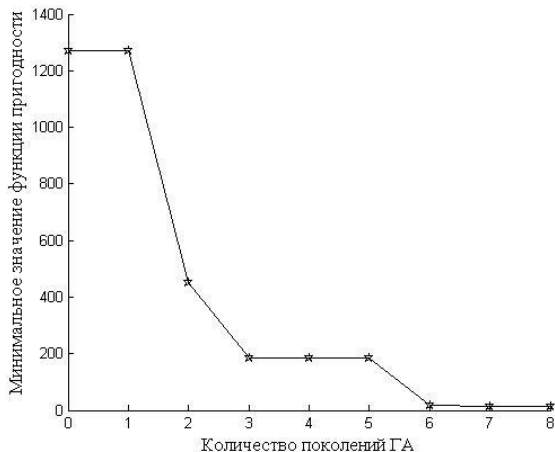


Рис. 2. Сходимость КОМБИ-ГА при 50 входных переменных и 20 процентах шума, используя ГА с операторами (оОК+3тОМ) (☆)

**Прикладная задача с реальными данными: моделирование уровней способностей игроков компьютерной игры [21].** Эта задача решалась на основе реального набора данных, взятого из *UCI Machine Learning Repository* [20].

В играх со стратегией в реальном времени (*Real-time strategy (RTS) games*) игроки разрабатывают фрагменты игры с конечной целью уничтожить штаб-квартиру своего противника.

В задаче рассматривается два типа переменных:

- показатели использования горячих клавиш (*hotkey usage*). Игроки могут настраивать интерфейс для быстрого выбора и контроля своих фрагментов, вручную разгружая некоторые аспекты игры, нажав на определенные иконки игрового интерфейса.

- характеристики цикла восприятия–действия (*perception–action–cycle*). Каждая переменная относится к периоду времени, когда игроки фиксируются и действуют в конкретном месте. Поэтому многие из этих переменных будут отображать процессы как восприятия, так и сосредоточения внимания, а также когнитивно-моторную скорость. Задача содержит 18 входных аргументов (целых и непрерывных).

Исходное значение – индекс лиги (*League Index*), кодирующий целыми значениями от единицы до семи уровень навыков игрока: бронзовый, серебряный, золотой, платиновый, специалист, высший специалист, профессионал. Использовалось 55 наблюдений. Оптимальная структура модели, найденная с помощью алгоритма КОМБИ, содержит только шесть значимых аргументов из всех 18, а именно  $x_1, x_7, x_9, x_{11}, x_{16}, x_{17}$ .

Структура лучшей модели, построенной по КОМБИ-ГА с использованием оОК, содержит такой же набор значимых аргументов, но в ней появляются три дополнительных аргумента с незначимыми параметрами. Гибрид КОМБИ-ГА сходится в среднем за 0,1 с. с использованием любого из рассматриваемых операторов, в то время как КОМБИ работает 31,5 с. График процесса конвергенции с оОК изображен на рис. 3.

При этом КОМБИ-ГА достаточно точно аппроксимирует исходное значение на всех трех подвыборках, что проиллюстрировано на рис. 4. Среднеквадратические ошибки выхода модели на подвыборках *A, B* и *C* равнялись, соответственно, 0,14, 0,26 и 0,53.

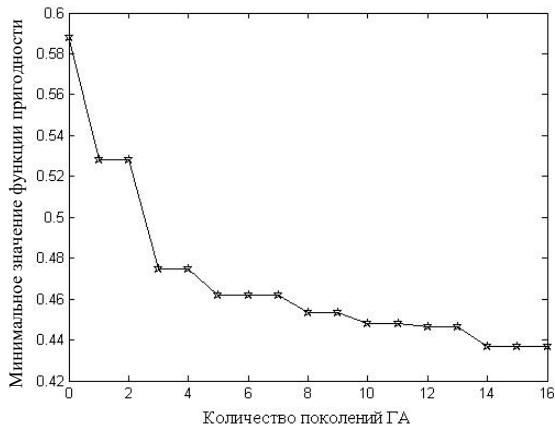


Рис. 3. Сходимость КОМБИ-ГА с использованием оОК

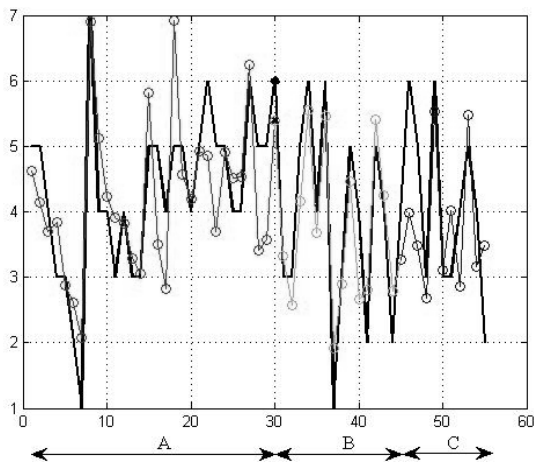


Рис. 4. Аппроксимация выходного значения на трех подвыборках – обучающей *A*, проверочной *B* и экзаменационной *C*

**Заключение.** Исследования, выполненные в работе, показывают, что гибридный переборный алгоритм КОМБИ-ГА позволяет эффективно найти истинную модель при большом количестве входных аргументов (>30). Надлежащий выбор операторов ГА существенно зависит от специфики и особенностей конкретной задачи. В частности, на тестовых данных наилучшие результаты были получены алгоритмом КОМБИ-ГА с использованием одноточечного оператора кроссинговера. Дальнейшие исследования целесообразно направить на совершенствование методики использования разработанного алгоритма путем автоматизации выбора целесообразных генетических операторов и их параметров для конкретных типов задач моделирования. Заслуживает также внимания задача разработки гибридных структур алгоритма КОМБИ с другими эволюционными алгоритмами.

1. Степашко В.С. Комбинаторная программа для структурной идентификации объектов и процессов управления / Под ред. А.Г. Ивахненко. – Киев: Техніка, 1980. – С.80–86.
2. Степашко В.С. Комбинаторный алгоритм МГУА с оптимальной схемой перебора моделей // Автоматика. – 1981. – № 3. – С. 31–36.
3. Ивахненко Г.А. Применение прогнозирующих алгоритмов МГУА для решения маркетинговых задач // Проблемы управления и информатики. – 2008. – № 2. – С. 149–152.
4. Chaudhuri R.A. Combinatorial GMDH Approach to Identification, Modelling and Prediction of Money Market State Variables // Int. Conf. on Technology and Business Management, 2012. – P. 208–221.
5. Kondryna A., Savchenko E., Direkorenko O. Forecasting of Ukraine Human Development Index by Combinatorial GMDH Algorithm // The 4th Int. Workshop on Inductive Modelling IWIM, 2011. – P. 46–50.
6. Samoilenko O. Combinatorial GMDH Algorithm Application to Predict the Dynamic of Demographic Characteristics in Ukraine // 4th Int. Conf. in Inductive Modelling, ICIM' 2013. – P. 312–314.
7. Савченко Е.А. Экспресс-прогноз уровня глюкозы в крови по комбинаторному алгоритму МГУА // УСиМ. – 2003. – № 2. – С. 107–112.
8. Обнаружение закономерностей взаимодействия ионов с поверхностью по комбинаторному алгоритму МГУА / А.Г. Ивахненко, Е.А. Савченко, Г.А. Ивахненко и др. // Проблемы управления и информатики. – 2003. – № 2. – С. 80–89.
9. Степашко В.С. Конечная селекционная процедура сокращения полного перебора моделей // Автоматика. – 1983. – № 4. – С. 84–88.
10. Самойленко О.А., Степашко В.С. Аналіз ефективності застосування частотного критерію в алгоритмі послідовного відсіювання неінформативних аргументів // Індуктивне моделювання складних систем: Зб. наук. праць. – К.: МННЦІТ та С НАНУ, 2012. – С. 191–209.
11. Єфіменко С.М. Комбінаторний алгоритм МГУА з послідовним ускладненням структур моделей на основі рекурентно-паралельних обчислень // Індуктивне моделювання складних систем: Зб. наук. праць. – К.: МННЦІТ та С НАНУ – 2014. – 6. – С. 81–89.
12. Stepashko V., Moroz O. Hybrid Searching GMDH-GA Algorithm for Solving Inductive Modeling Tasks // The 1th IEEE Int. Conf. on Data Stream Mining & Processing. – Lviv, Ukraine, 23–27 Aug., 2016. – Lviv Polytech. Nat. Univ., 2016 – С. 350–355.
13. Ивахненко А.Г., Степашко В.С. Помехоустойчивость моделирования. – Киев: Наук. думка, 1985. – 216 с.
14. Себер Дж. Линейный регрессионный анализ. – М.: Мир, 1980. – 456 с.

Окончание на стр. 88

15. *Holland J.H.* Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. – Univ. of Michigan, 1975. – 210 p.
16. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. – М.: ФИЗМАТЛИТ, 2006. – 320 с.
17. *Kaya Y., Uyar M., Tekin R.* A Novel Crossover Operator for Genetic Algorithms: Ring Crossover. Presented at CoRR, 2011.
18. *Глибовець М., Гуласєва Н.М.* Еволюційні алгоритми: Підручник. – К.: НаУКМА, 2013. – 828 с.
19. *Мороз О.Г.* Ефективність операторів мутації в задачі генетичного пошуку оптимальних моделей в перебі-

рному алгоритмі МГУА // Матеріали VIII Українсько-польської наук.-практ. конф. «Електроніка та інформаційні технології» (ЕЛІТ-2016), Львів–Чинадієво, 27–30 серп. 2016 р. – Львів: Вид. центр ЛНУ ім. Ів. Франка, 2016. – С. 331–332.

21. <http://archive.ics.uci.edu/ml/>
22. <https://archive.ics.uci.edu/ml/datasets/SkillCraft1+Master+Table+Dataset>

Поступила 23.11.2016  
Тел. для справок: +38 044 526-2549 (Київ)  
© О.Г. Мороз, 2016

UDC 681.5.015

О.Н. Moroz

### Sorting-Out the GMDH Algorithm with the Genetic Search of Optimal Model

**Keywords:** combinatorial GMDH algorithm (COMBI), genetic algorithm, hybrid GMDH-GA algorithm.

**Introduction.** Combinatorial GMDH algorithm (COMBI) is an effective means to solve the problems of structure-parametric identification, forecasting, building models of objects and processes from experimental data under uncertain conditions. But this algorithm is practically successful in solving such problems when the number of input variables (arguments) is less than 30 because of using exhaustive search of models with different structures from a given basic class of functions. Partly this problem of exponential growth of the algorithm complexity is solved by some algorithms with directed models search but they have their own limitations.

**Purpose.** The latest results concerning a sorting-out COMBI-GA hybrid algorithm with genetic search of optimal model as an alternative to known algorithms based on the determinate search procedures is summarized.

**Methods.** The genetic-based algorithm uses the genetic operators to find an optimal model and reduce the exhaustive search of the combinatorial algorithm.

**Results.** The effectiveness of four genetic operators is compared in COMBI-GA hybrid algorithm for solving test and real-world inductive modeling problems of diverse dimension.

**Conclusion.** Using GA is effective way to search for optimal model in sorting-out GMDH algorithms for quick solving inductive modelling tasks with large numbers of input variables (much more than 30).

