

ББК 32.973 я 2 6Ф7.3  
С74

**Справочник по типовым программам моделирования**  
С74 /А.Г. Ивахненко, Ю.В. Колпа, В.С. Степаненко и др.;  
Под. ред. А.Г. Ивахненко — К. : Техника, 1980.— 184 с, ил. —  
Библиогр.: - с.179—180.

В пер.. 85 к. 17 000 экз.

В справочнике описаны типовые программы моделирования на языках АЛГОЛ-60 и ФОРТРАН, предназначенные для решения задач технической кибернетики — распознания образов, прогнозирования, идентификации и оптимального управления сложными объектами в самых различных отраслях науки и техники. Расчитан на инженерно-технических работников, занимающихся моделированием сложных объектов и процессов на цифровых вычислительных машинах.

Рецензенты д-р техн. наук А.А. Молчанов, канд. техн. наук  
Б.В. Шеваршиձе.  
Редакция литературы по энергетике, электронике, кибернетике и связи  
Зав. редакцией З.В. Божко

## ВВЕДЕНИЕ

Если имеется достаточная и достоверная информация о всех компонентах сложного технологического, экономического, биологического или другого объекта управления, то синтез – прототипирующих и управляющих моделей выполняется известными детерминированными методами (например, методом имитационного моделирования). Справочник содержит описание и пакеты программ для тех случаев, когда такая информация отсутствует полностью или частично, а единственным источником информации служит небольшая таблица опытных данных. Эти программы реализуют индуктивный подход к решению задач моделирования при помощи перебора большого числа моделей-претендентов по целесообразно выбраным критериям. В результате удается получить единственную (для каждого вида критерия) модель оптимальной сложности.

Полученные модели описываются алгебраическими или конечно-разностными уравнениями и предназначаются для решения задач прогнозирования, автоматического управления с оптимизацией прогноза, распознавания образов и автоматической классификации.

Программы справочника можно применять раздельно или рассматривать как пакет программ для современного вычислительного комплекса.

Программы сгруппированы по типу решаемых задач в 1 – 5 главах Справочника. В начале каждой главы приводится общая характеристика данного класса задач и особенности помеченных программ.

- В описании каждой программы указываются:
  - язык программирования, тип транслятора и машины, тип задачи (прогнозирование, экстраполяция физического поля, идентификация открытого закона, синтез управления, оценка начальных или граничных условий, распознавание образов или классификация ситуации);
  - метод решения;
  - вид критерия селекции (регулярность, несмещенност, баланс переменных или комбинированный);

общий вид уравнения (алгебраические и гармонические модели или модели в виде конечно-разностных аналогов дифференциальных уравнений, вероятностные законы). Конечно-разностные модели имеют преимущество перед алгебраическими в смысле точности решений;

■ свойства объекта (стационарный или медленно изменяющий во времени);

■ используемые функции и подпрограммы;

■ комментарии к тексту и последовательность работы программы.

Для многорядных алгоритмов дополнительно указываются:

■ тип частных описаний — трехчленные  $q = a_0 + a_1x_1 + a_2x_2$ , четырехчленные  $q = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$ , шестичленные  $q = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2$ . В связи с широким применением линеаризации задачи при помощи преобразования переменных в последнее время применяются исключительно линейные трехчленные описания;

■ способ постепенного усложнения моделей, в многорядных алгоритмах (перебор всех пар аргументов, последовательный ввод пар с ортогонализацией, случайный перебор партнеров и т. п.);

■ способ выбора модели оптимальной сложности («с протекцией» или «без протекции переменным»).

Для пояснения принципиальных особенностей программ самоорганизации некоторые программы Справочника снабжены иллюстративными примерами. Подпрограммы, общие для нескольких программ Справочника, выпесены в приложение. Они могут применяться как самостоятельные либо в виде блоков.

В целом данный Справочник достаточно полно отражает современное состояние и прикладные возможности теории самоорганизации моделей па ЭВМ. Помеченные в нем программы являются эффективным инструментом для решения широкого круга задач моделирования сложных процессов и систем для целей прогноза и управления.

# Глава 1

## ИНДУКТИВНЫЙ МЕТОД САМООРГАНИЗАЦИИ МОДЕЛЕЙ НА ЭВМ

### 1. СИНТЕЗ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ПРИ ПОМОЩИ ИНДУКТИВНОГО МЕТОДА САМООРГАНИЗАЦИИ НА ЭВМ

Программы, представленные в Справочнику реализуют новый индуктивный метод самоорганизации моделей на ЭВМ, основанный на использовании опытных данных по так называемым внешним критериям.

Индуктивный метод моделирования, основанный на принципе самоорганизации, исходит из минимального объема требуемой для моделирования априорной информации. Однако это не означает, что достоверные знания об объекте должны скрываться от машинь: все, что известно точно, нужно ей сообщить. Недостаточное сведения машина находит при помощи перебора (последовательной оценки) большого числа вариантов моделей по некоторым внешним критериям. Таким образом, практический метод является комбинированным: используется достоверная априорная информация об объекте и индуктивный перебор и оценка моделей-претендентов. Оценки находятся по сравнительно небольшой части исходных опытных данных. Остальные данные используются для получения оценок коэффициентов и выяснения оптимальной структуры модели. Применение метода наименьших квадратов для оценки коэффициентов моделей не является принципиально необходимым. Известны алгоритмы, где оценки коэффициентов находятся также, как и структура модели оптимальной сложности, при помощи перебора вариантов дискретных знаний. Применение метода наименьших квадратов только уменьшает область перебора моделей.

Из сравнения метода регрессионного анализа и индуктивного метода самоорганизации моделей следует, что в каждом из них используется таблица опытных данных, однако, при регрессионном анализе приходится произвольно назначать структуру модели. Объясняется это тем, что критерий среднеквадратичной ошибки, рассчитанной по всем точкам таблицы опытных данных, является внутренним критерием. Любой внутренний критерий сравнения моделей приводит к ложному правилу: чем сложнее модель — тем она точнее. Сложность модели, например, полиномиальной, оценивается числом и высшей степенью слагаемых. Чем больше членов полинома — тем меньше среднеквадратичная ошибка.

В отличие от метода регрессионного анализа индуктивный метод самоорганизации моделей основан на применении внешних критерии\*\*. Выбора регулярности, минимума смещения, баланса переменных и комбинированных. Согласно теореме Гетея из математической логики о необходимости внешнего дополнения [13] только внешний критерий позволяет выбрать единственную (для каждого критерия) модель оптимальной сложности.

Структура модели оптимальной сложности соответствует минимуму внешнего критерия. Модель называется недостаточной, если сложность ее структуры используется в той же информации, тех же данных, которые используются и для получения самой модели.

\*\* Критерий называется внешним, если его определение основано на использовании той же информации, тех же данных, которые используются и для получения самой модели.

\*\* Критерий называется вспомогательным, если его определение основано на новой информации, «свежих» точках, не использованных при синтезе модели.

(оцениваемая, например, по числу членов и степени уравнения регрессии) меньше оптимальной. Модель называется пересложенной, если сложность ее структуры больше оптимальной. Переусложнение модели так же недопустимо, как и недостаточная ее сложность.

Есть еще одно существенное отличие регрессионного анализа и метода самоорганизации: регрессионный анализ позволяет строить модели только в области, где число коэффициентов модели равно или меньше числа точек таблицы опытных данных. Многорядные алгоритмы самоорганизации позволяют строить модели за пределы этой области. Число коэффициентов многорядной модели может измеряться миллионами, и все они будут оценены по таблице, имеющей, например,

всего 20 точек. Именно в этой новой области следует искать самые точные и несмещенные модели, так как сложность модели для сложных объектов принципиально необходима.

Регрессионный же анализ дает только сравнимую простые модели для оптимальной сложности.

Методы самоорганизации направлена на уменьшение необходимой априорной информации, вводимой человеком в машину.

В память машины вводят небольшую таблицу статистических

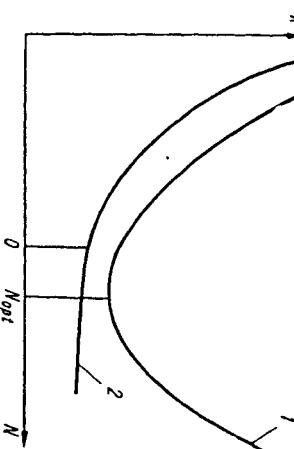


Рис. 1. Оптимизация длины таблицы опытных данных:  
1 — для нестационарных объектов; 2 — для объектов с постоянным законом

моделей по заданному критерию. В результате получается модель, дающая объективный прогноз.

Принцип самоорганизации состоит в том, что при постепенном усложнении моделей некоторые критерии (имеющие свойства «внешнего дополнения») проходят через свой минимум. Машина при помощи перебора моделей находит этот минимум и указывает единственную модель оптимальной сложности.

Величину минимального значения критерия — глубину минимума можно рассматривать как оценку качества моделирования. Если глубокий минимум не достигнут, значит, модель не найдена. Это может иметь место в следующих случаях: входные данные слишком зашумлены; не содержат существенных переменных, либо объект является индeterminированым (законы вообще нет); опорные функции не подходят для данного процесса; учитывается слишком мало западающих аргументов и т.п. В таких случаях необходимо расширить область перебора модели.

Необходимая глубина минимума оценивается на основании опыта моделирования данного класса объектов с учетом уровня помех. Если известен уже один из законов, действующих в объекте, то при помощи процедуры его восстановления по методу самоорганизации можно определить соответствующую глубину минимума. Такой же, ориентировочно, должна быть глубина и для других законов, действующих в объекте при тех же условиях.

Глубина минимума критерия выбора модели может быть использована для оценки всех параметров алгоритмов самоорганизации и, в том числе, для выбора оптимальной длины таблицы опытных данных. Длина таблицы постепенно наращивается, пока это увеличивает глубину минимума основного критерия. При этом возможны два результата (рис. 1): для нестационарных объектов и для объектов с постоянным законом.

\* Например, успеха в моделировании нельзя достичь при попытке синтезировать модель для прогноза выходной величины генератора случайных чисел по предыстории.

Первый результат свидетельствует об успешном определении оптимального числа точек предыстории (такой результат получается для нестационарных, медленно изменяющихся объектов), второй — характерен для стационарных объектов, в которых действует постоянный закон.

Выбор длины учитываемой предыстории функционально не связан с выбором шага дискретизации переменных во времени. Эта величина также должна выбираться с учетом угла наклона критерия.

Согласно теореме неполноты Геделя, задача идентификации структуры и параметров модели некорректна: только при наличии некоторого внешнего дополнения можно получить единственное ее решение. Во всех работах по моделированию, где тем или другим способом получена единственная модель, применено некоторое внешнее дополнение. Основной недостаток современного моделирования состоит в том, что внешнее дополнение выбирается неадекватно задаче.

## 2. ТРИ ОСНОВНЫЕ СТРУКТУРЫ АЛГОРИТМОВ САМООРГАНИЗАЦИИ МОДЕЛЕЙ НА ЭВМ

В самом общем виде алгоритм индуктивного метода самоорганизации моделей можно представить в виде общей схемы структуры алгоритмов самоорганизации моделей на ЭВМ (рис. 2), состоящей из генератора моделей ( $\Gamma M$ ), на выходе которого (в определенном порядке повышения сложности) получаются варианты (претенденты) моделей, и селектирующего устройства ( $K$ ), выбирающего (по заданному критерию) самую лучшую модель ( $B$ ).

В зависимости от устройства генератора моделей-претендентов можно различить три основные структуры алгоритмов самоорганизации: однорядные (комбинаторные) алгоритмы с заполнением коэффициентов, многорядные алгоритмы с пороговым отбором, многорядные алгоритмы с заполнением коэффициентов.

*Первый структурный — комбинаторный алгоритм МГУА* для малого числа переменных (до 20):

Рис. 2. Структурная схема алгоритмов самоорганизации моделей на ЭВМ:

$\Gamma M$  — генератор моделей-претендентов;  $K$  — критерий выбора;  $B$  — выбор единственної модели оптимальной сложности

получаем три укороченных, частных полинома:

$$q = a_0 + a_1 t, \quad q = a_0 + a_2 t^2, \quad q = a_1 t + a_2 t^2$$

и один полный полином. Таким образом, получаем таблицу полиномов с полным усложнением структуры (1). Каждый из полиномов должен быть оценен по избранному критерию селекции (2). Полином с наилучшим значением критерия является полиномиальной моделью оптимальной сложности (3). Производится пересчет оценок коэффициентов модели по всем точкам таблицы опытных данных (4).

Эта структура [19] реализована в так называемых «комбинаторных» алгоритмах МГУА и рекомендуется в тех случаях, когда число слагаемых полного полинома не превышает 20. При этом количество сравниваемых частных моделей равно 2<sup>20</sup>. «Заполнение» коэффициентов по очереди в разных комбинациях следует рассматривать как средство борьбы с помехами.

$a_0 + a_1 t$	$a_0 + a_2 t^2$	$a_0 + a_3 t^3$
$a_0 + a_1 t + a_2 t^2$		
$a_0 + a_1 t + a_2 t^2 + a_3 t^3$		

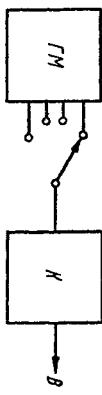


Рис. 2. Структурная схема алгоритмов самоорганизации моделей на ЭВМ:

$\Gamma M$  — генератор моделей-претендентов;  $K$  — критерий выбора;  $B$  — выбор единственної модели оптимальной сложности

Рис. 3. Первая структура (комбинаторный алгоритм МГУА) для малого числа переменных, до 20:

1 — таблица постепенного усложнения моделей; 2 — оценка коэффициентов по общему последовательности; 3 — выбор моделей по критерию комбинаторному критерию; 4 — пересчет оценок коэффициентов модели по всем точкам; 5 — выход

где  $F$  — свобода выбора (количество возможных частных моделей);  $S$  — номер ряда селекции.

Данная задача аналогична задаче определения оптимальной структуры пересчетного [17].

Оценки коэффициентов частных моделей рассчитываются по частям данных, называемых обучающей, последовательностью. Из всех частных моделей выбирается  $F$  лучших, дающих минимальные значения критерия селекции (пороговый самообзор лучших моделей).

Если обозначить число входных переменных первого ряда через  $t$ , то в большинстве программ выбирается  $F_1 = F_2 = \dots = F_r = l$  ( $r$  — номер последнего ряда селекции) \*

$$k = i + 1, \quad i + 2, \dots, l, \quad i = 1, 2, \dots, (l - 1), \quad S_1 = \frac{l(l - 1)}{2}.$$

В работе [38] рассмотрена задача определения оптимального закона изменения  $F$  для алгоритма МГУА.

$$F_S = f(S),$$

где  $F$  — свобода выбора (количество возможных частных моделей);  $S$  — номер ряда селекции.

В общем случае значения  $F_S$  ( $S = \overline{1, r}$ ) могут быть произвольными, удовлетворяющими ограничениям  $F_1 \leq t$ ,  $2 \leq F_S \leq C_{FS-1}^2$  ( $S = \overline{2, r}$ ).

\* В общем случае значения  $F_S$  ( $S = \overline{1, r}$ ) могут быть произвольными, удовлетворяющими ограничениям  $F_1 \leq t$ ,  $2 \leq F_S \leq C_{FS-1}^2$  ( $S = \overline{2, r}$ ).

### 3. НЕОБХОДИМОСТЬ СЛОЖНОСТИ МОДЕЛИ

Сложность модели определяется количеством аргументов и высшей степенью полинома. Для многих объектов модель оптимальной сложности лежит далеко за пределами линейных, квадратичных, кубических и других простых полиномиальных моделей. Сложность модели должна быть адекватной сложности моделируемого объекта. Сравним между собой методы синтеза моделей на основании анализа предыстории (т. е. по таблице опытных данных) — регрессионный анализ и алгоритмы самоорганизации модели (комбинаторные и многорядные):

**Регрессионный анализ.** При использовании регрессионного анализа приходится искать модели только в области, где число членов правой части уравнения не превосходит числа точек таблицы. Для устранения влияния случайных помех рекомендуется увеличить число точек в пять — десять раз. При десяти

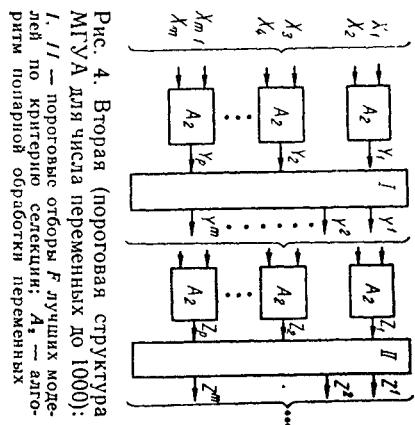


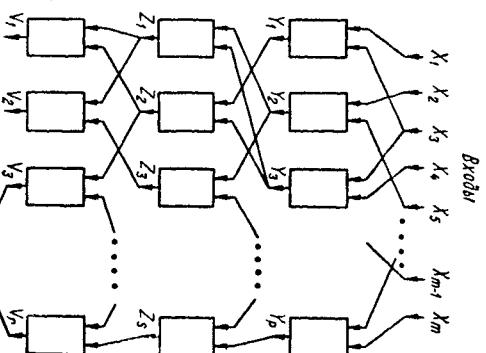
Рис. 4. Вторая (протotypная) структура МГУА для числа переменных до 1000:  
I, II — пороговые отборы  $F$  лучших моделей по критерию селекции;  $A_2$  — алгоритм поларной обработки переменных

Рис. 5. Третья структура МГУА (нейроноподобная сеть с приравниванием нулю коэффициентов для числа переменных до 100)

два вида многорядных алгоритмов: алгоритмы с протекцией определенным переменным, алгоритмы без протекции.

Допустим, в полном полиноме участвуют четыре переменных  $x_1x_2x_3x_4$ . Образуем матрицу сочетаний

	$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	$x_1^2$	$x_1x_2$	$x_1x_3$	$x_1x_4$
$x_2$		$x_2^2$	$x_2x_3$	$x_2x_4$
$x_3$			$x_3^2$	$x_3x_4$
$x_4$				$x_4^2$



Зависимость числа членов полного полинома от вида частного описания  
(на десятом ряду селекции)

Вид частного описания	Число членов полного полинома
$q_k = a_0 + a_1x_1 + a_2x_1^2$	$1 + 2^{10} + 2^{10} = 2049$
$q_k = a_0 + a_1x_1 + a_2x_1 + a_3x_1x_2$ $q_k = a_0 + a_1x_1 + a_2x_1 + a_3x_1x_2 +$ $+ a_4x_1^2 + a_5x_1^3$	$1 + 2^{10} + 2^{10} + 2 \cdot 2^2 \cdot 10 \approx 3 \cdot 10^6$

переменных полный степенной полином второй степени содержит: свободных членов — 1, линейных членов — 10, квадратов — 10, ковариаций второго порядка  $C_{10}^2 = 45$ . Всего: 66 членов

Даже если ограничиться второй степенью полинома, то потребуется таблица, содержащая  $66 \times 10 = 660$  опытных точек. Более сложные модели для регрессионного анализа практически недостижимы.

**Комбинаторные алгоритмы МГУА.** Практически на большей вычислительной машине (например, БЭСМ-6) можно перебрать все комбинации «зануления» коэффициентов полинома, содержащего 20—25 членов. Максимально допустимая сложность модели определяется этим чистом. Например, полное описание может иметь такой вид

$$q = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_5x_1^2 + a_6x_2^2 + a_7x_3^2 + a_8x_4^2 + a_9x_1x_2 + a_{10}x_1x_3 + a_{11}x_1x_4 + a_{12}x_2x_3 + a_{13}x_2x_4 + a_{14}x_3x_4 + a_{15}x_2^2x_1 + a_{16}x_2^2x_3 + a_{17}x_2^2x_4 + a_{18}x_3^2x_2 + a_{19}x_1x_2x_3 +$$

Три аргумента, третья степень полинома, двадцать слагаемых, число генерируемых при помощи «зануления» моделей  $(2^{20} - 1) = 1048575$ .

Применяя комбинаторные алгоритмы МГУА, следует помнить, что область более сложных моделей не может быть исследована и что именно там могут быть наиболее точные модели. Поэтому комбинаторные алгоритмы МГУА следует применять только там, где априори известны достаточны: в остальных случаях следует применять многорядные алгоритмы

**Многорядные алгоритмы.** При использовании нелинейного частного описания многорядные селекционные алгоритмы позволяют получать модели практически любой сложности (степень полинома на каждом ряду удваивается). При линейных

$$q = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_3$$

Эту структуру можно применить практически, когда в задаче содержится не более 100 исходных переменных.

частных описаниях степень полинома не увеличивается, меняется только число входящих в окончательный полином членов. Максимальное число членов модели при лестниче рядах селекции приведено в табл. 1.

Применение таких сложных моделей вполне возможно, если пользоваться системой частных описаний как системой уравнений. Исключение промежуточных переменных при линейных частных описаниях практически невозможно — запись слишком сложных полиномов возможна только в виде программы ЭВМ. При линейных частных описаниях исключение промежуточных переменных не вызывает затруднений.

*Многорядные алгоритмы с линеаризацией полного описания при помощи переборивания переменных.* Если задан полный полином максимальной сложности, состоящий из суммы новых и линейных членов (до 1000 слагаемых) и каждой член обозначен новой переменной, получаем линейный полином. Для определения его структуры и коэффициентов используется многорядный алгоритм с линейными частными описаниями, что освобождает от необходимости выбора вида частных описаний. С точки зрения допустимой максимальной сложности этот алгоритм занимает промежуточное положение между комбинаторным и многорядным алгоритмами.

#### 4. ОПТИМАЛЬНОЕ РАЗДЕЛЕНИЕ ТАБЛИЦЫ ОПЫТНЫХ ДАННЫХ

В алгоритмах самоорганизации моделей на ЭВМ таблица исходных данных делится по крайней мере на две части, называемые обучающей ( $A$ ) и проверочной ( $B$ ) последовательностями. При этом среднеквадратическая ошибка  $\Delta$  ( $B$ ), определяемая на проверочной последовательности (критерий регулярности), является одним из критериев для выбора структуры модели, синтезируемой по данным обучающей последовательности. Так создается первое внешнее дополнение.

Если ставится задача оптимизации разделения данных на обучающую и проверочную части, то, согласно теореме Геделя, требуется еще одно, второе, внешнее дополнение. Таблицу данных приходится делить на три части: обучающую, проверочную и две экзаменационные, последовательности ( $A$ ,  $B$ , и  $C$ ,  $D$ ). Ошибки  $\Delta$  ( $C$ ), определяемая на экзаменационных данных, может служить вторым внешним дополнением, поэтому таблицу следует разделить так, чтобы  $\Delta(C) \rightarrow \text{тип}$ .

Чтобы оптимизировать разделение данных на три последовательности, требуется еще одно, третье, внешнее дополнение  $\Delta(D)$ . Для точного соблюдения гипотезы Геделя необходимо, чтобы  $\Delta(D) \rightarrow \text{тип}$  и  $\Delta$ .

В программах часто выбираются следующие разделения опытных данных.

Для критерия регулярности:

$$N_A = 0.7N; N_B = 0.3N; N_C = 0, N_D = 0,$$

$$N_A = 0.6N; N_B = 0.2N; N_C = 0.1N, N_D = 0.1N.$$

Для критерия минимума смещения:

$$N_A = 0.5N; N_B = 0.5N, N_C = 0, N_D = 0;$$

$$N_A = 0.4N; N_B = 0.4N; N_C = 0.1N, N_D = 0.1N,$$

где  $N_A$ ,  $N_B$ ,  $N_C$ ,  $N_D$  — число точек обучающей, проверочной, первой и второй экзаменационных последовательностей соответственно;  $N$  — общее число точек.

Более точно оптимальное разделение находится по второму внешнему критерию качества разделения.

В качестве второго внешнего дополнения могут применяться также критерии минимума числа рядов селекции и минимаксный.

*Критерий минимума числа рядов селекции.* Согласно этому критерию выбирается такое разделение таблицы опытных данных, при котором достигается минимум числа рядов многорядного алгоритма МГУА, и, следовательно, получается самая простая модель.

Было обнаружено существование оптимума числа рядов многорядного селективного алгоритма МГУА при варьировании соотношения числа точек обучающей последовательности  $N_A$  и числа точек проверочной последовательности  $N_B$ . На рис. 6 показан график оптимизации числа рядов селекции для алгоритмов МГУА второй структуры. Точки ранжируются по дисперсии в один ряд, а затем строится парабола. Минимум числа рядов указывает оптимальное разделение, так как, чем меньше рядов, тем выше (и достовернее) модель.

Таким образом, в качестве второго внешнего дополнения можно использовать число рядов селекции.

*Минимаксный критерий.* Выбирается такое разделение таблицы, при котором достигается минимум наибольшего значения критерия несмещенности  $n_{\max} \rightarrow \text{тип}$ . Этот критерий обладает повышенной помехоустойчивостью [21].

## Глава 2

### ВЫДЕЛЕНИЕ ТРЕНДОВ

#### ДЛЯ ДОЛГОСРОЧНОГО ПРОГНОЗИРОВАНИЯ

##### 1. ЗАДАЧА ДОЛГОСРОЧНОГО ПРОГНОЗИРОВАНИЯ ПО ТРЕНДАМ

Методика долгосрочного прогнозирования сложных процессов, согласно современным представлениям, должна предусматривать использование развитого вычислительного комплекса машин с графическим дисплеем [26], работающим в режиме диалога человека-машин.

Методы самоорганизации прогнозирующих моделей на ЭВМ наиболее перспективны, хотя будущий диалоговый комплекс не должен исключать использования других, а также комбинированных методов (например, методов имитационного, или динамического, моделирования).

*Прогнострирующей моделью* называют уравнение (или систему уравнений), описывающее предполагаемый ход моделируемого процесса в будущем. Соответствующие уравнениям зависимости, представленные в виде таблиц или графиков, являются прогнозом.

При использовании методов самоорганизации прогнозирующие модели строятся на основе опытных данных (по предварительным наблюдениям процесса или, иначе, по его предыстории). Обычно такая информация представляется в таблицах через равные отрезки времени (дискретные интервалы или шаги). Прогноз с временным упреждением на один-два интервала вперед называется краткосрочным, на три — среднесрочным, а на десять и более — долгосрочным. Задача долгосрочного прогнозирования процесса (или временного ряда) является определение его систематической составляющей, называемой *тенденцией*. Общие тенденции изменения временных рядов могут иметь линейный (например, возрастанию) или циклический характер и определяться, соответственно, как полиномиальный, или, так и циклический, предикт. В ряде случаев тренд может являться функцией не только времени, но и ряда других побочных переменных и параметров (при этом он применяется для построения так называемых нормативных прогнозов). Флуктуации и перегуляризующие временные рядов включают в себя ошибки измерения, а также отклонения от тренда, вызываемые воздействием неучитываемых изменических факторов.

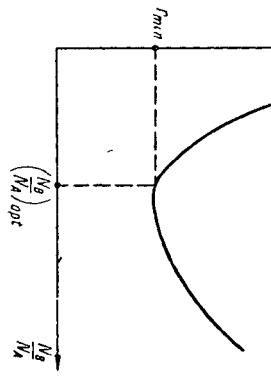


Рис. 6. Оптимизация числа рядов селекции для алгоритмов МГУА второй структуры

Приведенные ниже четыре вычислительные программы для выделения трендов по зашумленным опытным данным можно рассматривать как небольшой пакет проблемно-ориентированных программ, предназначенных для решения некоторых задач прогнозирования. Общий для этих программ является то, что они используют функции определенного вида, что обуславливает некоторую их ограниченность.

Альтернативой может быть прогнозирование на основе автогрессионных или более сложных коечечно-разностных уравнений. При синтезе трендов оптимальной сложности организуется перебор большого числа трендов-претендентов по критериям регуляризации, минимума смещения, критерии сходимости баланса переменных или комбинированым критериям.

**Критерий регуляризации.** Проверочная последовательность используется в МГУА для выбора числа членов и степени уравнения регрессии; это позволяет получить наиболее регулярные решения (оптимальной сложности), мало чувствительное к небольшим изменениям исходных данных. Такое решение дает небольшую ошибку на новых точках, в чем и состоит основная цель МГУА в задачах однократного (краткосрочного) прогноза и, иногда, идентификации. Величина критериев регуляризации вычисляется по данным отдельной проверочной последовательности.

Разделение исходной выборки на две последовательности — обучающую и проверочную — производится по одному из следующих приемов [14]:

- Опытные точки ранжируются в ряд по дисперсии (квадрат расстояния от центра «тяжести») пространственной фигуры, охватывающей все представляющие точки):

$$D_i^2 = \sum_{i=1}^n (x_{ij} - \bar{x}_i)^2, \quad (2.1)$$

где  $j = 1, 2, \dots, N$ ;  $N$  — общее число точек;  $n$  — число аргументов;

$$\bar{x}_i = \frac{1}{N} \sum_{j=1}^N x_{ij}.$$

В этом случае для определения коэффициентов частных описаний используются возможностью удалить точки.

После ранжировки по дисперсии можно произвести дополнительное разделение точек на нечетные — четные. В таком случае первая и вторая части последовательности имеют близкие статистические свойства.

Если исходные данные снимаются в строгой временной последовательности (например, опытные данные физического поля или технологического процесса), то ранжировка точек в ряд недопустима. В этом случае можно применить деление точек по четным и нечетным индексам. После разделения по тому или другому способу первые  $N_A$  точек составляют обучающую последовательность, следующие  $N_B$  точек — проверочную.

В качестве критериев регуляризации используются:

- относительная среднеквадратичная ошибка

$$\Delta^2(B) = \frac{\sum_{i=1}^{N_B} (q_i^* - q_i)^2}{\sum_{i=1}^{N_B} (q_i^*)^2} \rightarrow \min; \quad (2.2)$$

коэффициент корреляции

$$\kappa_{qq}^* = \frac{\sum_{i=1}^{N_B} q_i^* q_i}{\sqrt{\sum_{i=1}^{N_B} (q_i^*)^2 \sum_{i=1}^{N_B} q_i^2}} \rightarrow 1; \quad (2.3)$$

#### индекс корреляции

$$\rho = \sqrt{1 - \frac{\sum_{i=1}^{N_B} (q_i^* - q_i)^2}{\sum_{i=1}^{N_B} (q_i^* - \bar{q}_i^*)^2}} \rightarrow 1, \quad (2.4)$$

где  $\Delta^2(B)$  — относительная среднеквадратичная ошибка,  $q_i$ ,  $q_i^*$  — полученные по модели и действительное значение выходной величины в  $i$ -й точке ( $i = 1, 2, \dots, N_B$ ),  $N_B$  — число точек проверочной последовательности;

$$\bar{q}_i^* = \frac{1}{N_B} \sum_{i=1}^{N_B} q_i^*.$$

(Величины  $\Delta^2(B)$ ,  $\kappa_{qq}^*$ ,  $\rho$  вычисляются на проверочной последовательности  $B$ ).

Чем меньше величина относительной среднеквадратичной ошибки или чем больше величина коэффициента корреляции или индекса корреляции, тем модель более регулярна.

Значения коэффициента корреляции и индекса корреляции в меньшей степени зависят от конкретного процесса и более подходят для сравнения результатов моделирования, чем значение среднеквадратической ошибки на проверочной последовательности.

**Критерий минимума смещения.** Модели, найденные по минимуму критерия регуляризации, хорошо решают задачу краткосрочного прогноза при малом уровне шумов в исходных данных. При увеличении помех минимум критерия регуляризации смещается влево, в сторону выбора более простых моделей, и при большой интенсивности шумов оптимальными оказываются линейные прогнозирующие модели.

Более устойчивыми к помехам являются различные формы критерия минимума смещения, так как, хотя при увеличении помех минимальное значение показателя смещения тоже увеличивается, минимум получается при одинаковой сложности модели, отвечающей истинной модели (это объясняет, почему такой критерий рекомендуется для решения задачи открытия законов). При малых шумах в исходных данных критерий регуляризации и критерий несмещенности равносочены и приводят обычно к одинаковым результатам.

Общая, основная идея построения критерия минимума смещения состоит в том, что модели, получаемые при использовании различных частей таблицы исходных данных, должны по возможности мало отличаться друг от друга, а при полном отсутствии смещения — совпадать. Это означает, что если исходные данные разбиты на две части,  $A$  и  $B$ , полученная на множестве точек  $B$ , должна возможна меньше отличаться от модели, полученной на множестве точек  $A$ .

**Критерий минимума смещения, основанный на анализе решений.** Для расчета показателя смещения все имеющиеся экспериментальные данные делятся на две части  $A$  и  $B$ . Сначала первая последовательность данных является обучающей, а вторая проверочной (частные модели), получаемые при этом, обозначим  $q(A) = f(x_i, x_j)$ . Далее первая последовательность служит проверочной, а вторая — обучающей, что выражается уравнением регрессии (частной моделью)  $q(B) = f(x_i, x_j)$ . Близость этих уравнений оценивается по величине среднеквадратичного расхождения их выходов, рассчитанных по всем таблицам точкам:

$$n_{cm}^2 = \frac{\sum_{i=1}^N |q_i(A) - q_i(B)|^2}{\sum_{i=1}^N q_{i, \text{табл}}^2}. \quad (2.5)$$

В многорядных алгоритмах МГУА эта величина, называемая показателем смещения, используется для выбора  $k$  лучших частных описаний, пропускаемых на последующие ряды селекции. В качестве суммарной характеристики смещения частных моделей ряда используется среднее значение показателей смещения самых несмешанных уравнений.

$$N_{\text{см}} = \frac{1}{F} \sum_{i=1}^F n_{\text{см},i}^2, \quad (2.6)$$

В некоторых программах вместо  $N_{\text{см}}$  используется одно, лучшее значение  $n_{\text{см},i}^2$ . Ряды селекции наращиваются до тех пор, пока величина  $N_{\text{см}}$  уменьшается в комбинированных алгоритмах МГУА. Значения  $n_{\text{см}}^2$  рассчитываются для всех моделей, заданных в таблице постепенного усложнения, и в результате выбирается самая несмешанная модель.

В некоторых алгоритмах применяются следующие разновидности показателя смещения, основанные на анализе решений:

$$n_{\text{см}}^2 = \frac{1}{N} \sum_{i=1}^N [q_i(A) - q_i(B)]^2 \rightarrow 0, \quad (2.7)$$

$$n_{\text{см}}^2 = 2 \frac{\sum_{i=1}^N q_i(A) q_i(B)}{\sum_{i=1}^N (q_i^2(A) + q_i^2(B))} \rightarrow 1. \quad (2.8)$$

*Критерий минимума смещения, основанный на анализе коэффициентов*. В том случае, когда все образом нормированные аргументы частных моделей МГУА имеют примерно одинаковые максимальные и минимальные значения, для оценивания частных описаний можно воспользоваться критерием минимума смещения коэффициентов

$$n_{\text{см}}^2 = \frac{\sum_{i=1}^{\xi} (a_i - b_i)^2}{\sum_{i=1}^{\xi} (a_i^2 + b_i^2)}, \quad (2.9)$$

где  $\xi$  — общее число параметров модели,  $a_i$  — оценки коэффициентов, полученные на последовательности  $A$ , а  $b_i$  — на  $B$ . Возможны и другие формы этого критерия, однако соответствующий показатель смещения, как функция параметров модели, должен быть безразмерным, симметричным и малочувствительным к различию в малых коэффициентах, например,

$$n_{\text{см}}^2 = 2 \frac{\sum_{i=1}^{\xi} a_i b_i}{\sum_{i=1}^{\xi} (a_i + b_i)^2} \rightarrow 1. \quad (2.10)$$

В ортогонализованных алгоритмах МГУА критерий минимума смещения коэффициентов упраздняется. Действительно, в этом случае частные модели содержат всего один коэффициент, смещение которого может быть вычислено по формуле

$$n_{\text{см},1}^2 = \frac{(a - b)^2}{a^2 + b^2} \quad (2.11)$$

или

$$n_{\text{см},1} = \frac{|a - b|}{|a| + |b|}. \quad (2.12)$$

*Критерий минимума смещения, использующий табличные суммы и разности*.

Достоверность оценок параметров моделей, как известно, пропорциональна  $\sqrt{N}$ . Поэтому деление таблицы исходных данных на две части снижает достоверность коэффициентов  $a_i$  и  $b_i$  и, следовательно, достоверность самого показателя  $n_{\text{см}}$ .

Для устранения этого в последние время разработаны критерии, не требующие разбиения таблицы данных на множества  $A$  и  $B$  [15, 21]. При этом первый раз оценки параметров моделей  $A_i$  находятся по всем точкам, а в качестве новой информации используются табличные суммы или табличные разности, которые позволяют определить, соответственно,  $A_{\text{лин}}$  или  $A_{\text{диф}}$ . Показатели смещения, построенные на основе этих оценок получили название «интегрального» и «дифференциального», они могут использоваться, как в критерии, основанном на анализе разновидности, так и в критерии смещения коэффициентов. Предпочтение отдается первому.

В «интегральных» критериях несмешанности матрицы нормальных уравнений Гаусса для нахождения  $A_{\text{лин}}$  составляются так же, как и в обычном регрессионном анализе, то есть на табличных суммах значений аргументов и функций. Для критерия несмешанности в «дифференциальной» форме процедура оценивания более сложная [13].

*Критерий минимума смещения для динамических моделей*. В случае динамических моделей, кроме описанных выше критерев, применяют еще один специфический показатель смещения, который оценивается разница в результатах пошагового многократного интегрирования уравнения динамики с коэффициентами  $a_i$  и  $b_i$ :

$$n_{\text{см}}^2 = \frac{\sum_{k=1}^N |X_A(k, x_0) - X_B(k, x_0)|^2}{\sum_{k=1}^N X_{\text{крабол}}^2}, \quad (2.13)$$

где  $X_A(k, x_0)$  и  $X_B(k, x_0)$  — интегральные кривые модели с параметрами, определенными на  $A$  и  $B$ . Легко можно построить «интегральную» и «дифференциальную» формы этого критерия, если найти интегральные кривые модели, отвечающие опискам параметров  $a_{\text{инт}}$  и  $a_{\text{диф}}$ .

В комбинаторных алгоритмах МГУА применение этого критерия не вызывает осложнений: для многорядных селективных процедур такой показатель смещения оказывается весьма неудобным и поэтому рекомендуется только для окончательного выбора единственной модели из группы наилучших моделей, для которых получено развернутое описание.

*Критерий минимума смещения для коротких выборок*. В случае очень коротких рядов наблюдений, когда деление таблицы данных на две части нецелесообразно (или даже невозможно), кроме «интегральной» и «дифференциальной» форм критерия минимума смещения можно использовать показатель смещения, при котором оценки параметров модели один раз находят по методу наименьших квадратов (МНК), а второй раз — по методу ортогональной регрессии [30]; причем в обоих случаях используется полная таблица данных.

Необходимые расчетные формулы приведены в работе [6]. Для динамических моделей применение такого критерия является обоснованным с точки зрения математической статистики.

*Критерий, использующий экстремальные значения показателя смещения*. Самая несмешанная модель должна давать по возможности неизменное и малое значение показателя смещения при всех способах разделения точек на множества

*A* и *B*. Поэтому  $n_{\text{см}}^2$  \* служит более совершенной мерой смещенностя, чем одиничный показатель  $n_{\text{см}}^2$ , определенный для одного способа разделения точек.

В работе [6] описан алгоритм приближенного определения  $n_{\text{см}}^2$  max, не требующий перебора всех вариантов деления таблицы и заключающийся в последовательном переводе в множество *B* точек с наибольшей величиной «индекса точки», определяемого по рекуррентным формулам. С помощью этих алгоритмов можно достаточно точно и экономично вычислительной точки зрения получить оценку  $n_{\text{см}}^2$  max.

Критерий рекомендуется для временных рядов умеренной длины. Помехустойчивые формы критерия минимума смещения. Как известно, модели, построенные на множествах *A* и *B*, достаточно хорошо аппроксимируют различные значения функций на участке интерполяции, и поэтому различия моделей в показателе  $n_{\text{см}}^2$  незначительны. При определенном уровне шума может выбираться модель  $n_{\text{см}}^2$  незначительна. Для повышения помехустойчивости критерия, т. е. повышения того уровня шумов, при котором выбор истинной модели не нарушается, необходимо выйти за пределы участка интерполяции при подсчете показателя  $n_{\text{см}}^2$  [17]. При этом в приведенных выше формулах (2.5) — (2.13) суммирование производится до  $\alpha N$ , где  $\alpha \approx 1.5 \dots 2.5$ , а в знаменателях формул вместо  $N$  —  $q_{k\text{раб}}^2$ , используются сумма  $\sum_{k=1}^N (q_k^2(A) + q_k^2(B))$  либо  $\alpha \sum_{k=1}^N q_{k\text{раб}}^2$ .

Для прогнозирующих моделей в виде трендов и динамических (конечно-разностных) моделей выход за пределы участка интерполяции практически труда не представляет. Для статических объектов существует ряд искусственных приемов, расширяющих область пространства значений аргументов, в которой оценивается качество модели [17, 25].

**Критерий сходимости пошагового интегрирования конечно-разностных моделей.** Критерий представляет собой среднеквадратическую ошибку пошагового интегрирования и вычисляется по формуле

$$t^2(N) = \frac{\sum_{k=1}^N (q_k - q_{k\text{раб}})^2}{\sum_{k=1}^N q_{k\text{раб}}^2} \rightarrow \min \quad (2.14)$$

на интервале интерполяции (в области, где заданы опытные точки).

Критерий рекомендуется для конечно-разностных прогнозирующих моделей авторегрессионного типа  $q_{k+1} = a_0 + a_1 q_k + a_2 q_{k-1} + a_3 q_{k-2} + \dots$ . Он не требует разделения данных на обучающую и проверочную последовательности: проверку точности пошаговой интегрированием конечно-разностных моделей можно выполнить на тех же точках, по которым синтезирована сама модель (множество точек  $N = A + B$ ). Объясняется это тем, что интеграл и его аргумент линейно независимы, поэтому интеграл рассматривается как новая, свежая информация, и критерий  $t^2(N)$  вычисляется на участке интерполяции, т. е. в области, где заданы точки опытных данных.

Критерий баланса переменных и его применение для синтеза моделей. Основная идея критерия баланса переменных состоит в следующем: на интервале интерполяции, где заданы экспериментальные точки, устанавливается некоторая математическая связь между переменными, образующими систему. Критерий баланса переменных требует, чтобы эта связь, называемая также законом, выполнялась и на интервале экстраполации, т. е. в области прогнозов. Если такой закон действительно существует и остается постоянным, то критерий баланса переменных обеспечивает точные долгосрочные прогнозы.

\* Наибольшее значение  $n_{\text{см}}^2$  при всех возможных вариантах деления таблицы дано на две части

Пусть  $f(q_1(t), q_2(t), \dots, q_S(t)) = 0$  — функция баланса (т. е. «закона», связывающей переменные  $q_1(t), q_2(t), \dots, q_S(t)$ ). Из множества всех прогнозирующих моделей для переменных  $q_i(t)$  нужно выбрать такую систему, для которой на интервале экстраполяции (в районе точки прогноза) эта связь выполняется наилучшим образом. Нарушение баланса переменных можно характеризовать величинами небаланса

$$b_i = f(q_1(t_i), q_2(t_i), \dots, q_S(t_i)), \quad (2.15)$$

которые рассчитываются для  $t_i$ , принадлежащих интервалу экстраполяции, по различным прогнозирующими моделям для каждой переменной  $q_i(t)$ . В результате выбирается наилучшая комбинация моделей. Их усложнение уdoneнее всего задавать в виде специальных таблиц постепенного усложнения [14, 19].

Критерий баланса позволяет выбрать наилучший прогноз системы переменных из набора так называемых «вееров» прогнозов (трендов) для каждой переменной. В некоторых задачах функция связи между переменными ясна из физических соображений и использование ее в алгоритме является вполне естественным. В других случаях из физики процесса можно судить о существовании функциональной зависимости между переменными, однако вид ее неизвестен. При этом рекомендуется предварительно восстановить функцию баланса, используя один из алгоритмов МГУ на основе критерия минимума смещения.

В том случае, когда прогнозируемое переменными признали несвязанными, возможно искусственное расширение множества переменных за счет дополнения их временными рядами, образованными из исходных по выбранным законам. Эти зависимости и высступают в дальнейшем в качестве функций баланса. При этом линейные функции баланса приводят к неоднозначности выбора модели, в то время как нелинейные — обеспечивают более надежную селекцию.

Если взаимосвязь между переменными существует, то на участке интерполяции могут быть построены несмешенные законы:

$$\begin{aligned} q_1 &= f_1(q_2, q_3, \dots, q_S), \\ q_2 &= f_2(q_1, q_3, \dots, q_S), \\ &\vdots \\ q_S &= f_S(q_1, q_2, \dots, q_{S-1}). \end{aligned} \quad (2.16)$$

Допустим, что функции (2.16), называемые прямыми, достаточно просты и допускают «обращение», т. е. выражение некоторых аргументов, входящих в правую часть, через остальные. Тогда из системы уравнений (2.16) можно получить

$$\begin{aligned} q_1 &= f_{1\text{обр}}(q_2, q_3, \dots, q_S), \\ &\vdots \\ q_S &= f_{S\text{обр}}(q_1, q_2, \dots, q_{S-1}), \end{aligned} \quad (2.17)$$

причем  $i$ -я обратная функция находится из  $i$ -й прямой ( $i \neq 1$ ). Из всех возможных вариантов построения обратной функции следует предпочтеть тот, при котором она находится из самой несмешенной прямой (исходной) функции. Величины  $b_{1i} = (f_1 - f_{1\text{обр}})^2, b_{2i} = (f_2 - f_{2\text{обр}})^2, \dots, b_{Si} = (f_S - f_{S\text{обр}})^2$ , рассчитанные на интервале экстраполяции ( $t_i$  принадлежит интервалу прогнозирования) по прогнозирующими моделям (трендам) для каждой переменной, характеризуют небаланс системы прогнозов в точке, а величина

$$B = \sum_{i=1}^m \left( \frac{\sum b_{ti}}{\sum f_i^2(t)} \right) \quad (2.18)$$

используется для выбора системы моделей.

*Практические рекомендации по выбору критерия баланса.* Среди комбинаций трендов (по одному из «квир») каждой переменной), дающих малое, близкое к нулю значение критерия баланса переменных, обязательно находится одна комбинация, отвечающая действительному ходу процесса, так как закон, связывающий переменные, постоянен во времени. Однако среди таких комбинаций могут быть и ложные, не отвечающие ходу процесса, поэтому необходимо выделить «истинный» баланс среди нескольких «ложных». Для решения этой задачи рекомендуется использовать дополнительные ограничения (критерий «физической реализуемости прогноза» и селекцию моделей по двум целесообразно выбранным критериям).

При реализации критерия баланса в виде «скользяя разности прямых и обратных функций» необходимо иметь в виду, что в случае тождественности  $f_i$  и  $f_{\text{обрат}}(i)$  величина  $B$  не позволяет сделать однозначный выбор модели, поэтому рекомендуется переход к показателю

$$B^N = \sum_{i=1}^m \left( \frac{\sum [(q_i(i) - f_i(i))^2 + (q_i(i) - f_{\text{обрат}}(i))^2]}{\sum_i x_j^2(i)} \right). \quad (2.19)$$

В случае дифференциальных прогнозирующих моделей выход за пределы интервала интегрирования при формировании критерия баланса осуществляется при многошаговом интегрировании уравнений динамики. Наиболее точные прогнозы получаются при использовании в критерии истинных функций баланса, определенных из физических соображений.

**Комбинированный критерий селекции, его разновидности и свойства.** Выбор критерия отбора (селекции) определяется типом решаемой задачи, принадлежит автору модели и отражает его цели. По рекомендациям, приведенным в работе [3], можно целесообразно выбрать один из эвристических критерии. Однако в некоторых ситуациях требования к моделям оказываются противоречивыми и не могут быть выражены в виде одного минимизируемого критерия. Кроме того, иногда выбранный критерий не обеспечивает единственностю выбора структуры модели или же требует слишком больших вычислений, так что желательно сокращение объема перевора. В ряде случаев может оказаться недостаточной «克莱вность» критерия, и требуется предварительно сократить число оцениваемых моделей, чтобы поместить в область надежной селекции.

В том случае, когда необходимо согласование нескольких противоречивых свойств (качеств) модели, рекомендуется перейти к комбинированным критериям [3], которые обычно формируются из составляющих критерии  $K(1)$  и  $K(2)$ :

$$K_{\text{комб}} = \sqrt{\lambda K_1^2 + (1 - \lambda) K_2^2}, \quad 0 < \lambda < 1. \quad (2.20)$$

и находить модели, неулучшаемые по обоим критериям, а при изменении  $\lambda$  во всем интервале [0, 1] построить множество паретоских структур моделей \*.

Кроме согласования противоречивых требований к модели (практически, задана многокритериальной оптимизацией), применение комбинированных критерии позволяет обеспечить единственность выбора структуры в том случае, когда один из критерии приводит к многозначности модели. Оба критерия должны быть выбраны целесообразно, т. е. с учетом физического смысла решаемой задачи моделирования. Модели, равновенные по одному из критерии, будут отличаться по другому, что и определяет однозначный выбор.

**Комбинированный критерий, учитывающий исходные критерии, может формироваться и в виде**

$$\sim K = \sqrt{\lambda \left( \frac{K_1}{K_{1\max}} \right)^2 + (1 - \lambda) \left( \frac{K_2}{K_{2\max}} \right)^2}, \quad 0 < \lambda < 1, \quad (2.21)$$

\* Паретоскими называются модели, удовлетворяющие условиям многокритериального выбора по противоречивым критериям. Принцип многокритериальной оптимизации сформулирован французским математиком Парето в конце XIX века.

приведенном к безразмерным величинам, что необходимо в том случае, когда  $K_1$  и  $K_2$  имеют различные единицы измерения и различный физический смысл.

*Последовательное использование (шерхария) критерия селекции* могут быть основными и вспомогательными. Основной критерий выбирается эвристически в зависимости от мотивации (цели) решения задачи моделирования. Вспомогательные критерии всегда подчинены основному и служат для сокращения перебора моделей. В качестве вспомогательных критерий используются критерии, имеющие плавные характеристики при изменении сложности модели, особенно если основной критерий такой характеристики не имеет.

Главность характеристики вспомогательного критерия позволяет рассматривать только часть моделей при переборе по таблицам усложнения функций (например, можно все модели предварительно отсеять по критерию регулярности, отбросить неудовлетворительные и перейти к селекции по основному критерию — минимума смещения).

Последовательное применение нескольких критерии селекции удобно заменить комбинированными критериями.

*Практические рекомендации по применению комбинированных критерии селекции.* Теория самоорганизации подчеркивает, что целесообразный выбор критерия селекции и порядка их использования принципиально отличается от выбора критерия регулярности,  $n_{\text{см}}$  — критерий минимума смещения,  $B_j$  — различия виды критерии баланса переменных,  $t^2$  — критерий устойчивости прогнозов.

Ориентировочно можно сформулировать следующие правила применения комбинированных критерии селекции:

1. Для алгебраических (полиномиальных) моделей, не содержащих среди аргументов время (т. е. для моделей, не предназначенных для долгосрочного прогноза), следует применять комбинированный критерий «смещение плюс регулярность»

$$K_1 = \sqrt{n_{\text{см}}^2 + \Delta^2(B)} \rightarrow \text{мин.} \quad (2.22)$$

2. Для гармонических и алгебраических моделей, использующих функции времени (т. е. предназначенных для долгосрочного прогноза), в случаях, когда известны и может быть открыт закон физического баланса переменных, рекомендуется комбинированных критерии селекции:

$$K_2 = \sqrt{n_{\text{см}}^2 + b^2 \rightarrow \text{мин.}} \quad (2.23)$$

3. Для дифференциальных моделей (в виде конечно-разностных уравнений), предназначенных для многократного долгосрочного прогноза при помощи пошагового интегрирования, рекомендуется комбинированный критерий «несмещение плюс устойчивость прогноза»

$$K_3 = \sqrt{n_{\text{см}}^2 + t^2(N)} \rightarrow \text{мин.} \quad (2.24)$$

При использовании дифференциальных моделей результаты долгосрочного прогноза могут быть неудовлетворительными, если вычислительная процедура не обладает достаточной устойчивостью, т. е. приводят к накоплению ошибок. Существует два способа достижения устойчивых результатов:

устредение некоторого числа лучших частных моделей на каждом ряду селекции или только в конце сингата модели.

Использование комбинированного критерия  $K_3$ , содержащего требования устойчивости. Величина  $t^2(N)$  формируется как сумма квадратов разностей таблицных значений выхода модели и рассчитанных в процессе многократного интегрирования уравнения динамики.

Алгоритмы с усреднением также позволяют получать более устойчивые модели, истинная же модель всегда устойчива.

Применяются также комбинированные критерии, учитывающие три или четырех составляющих, например, вида

$$K_4 = \sqrt{\frac{n_{cm}^2}{n_{cm}^2 + b^2 + r^2}} (N) \rightarrow \text{тп.} \quad (2.25)$$

#### Адаптация коэффициентов моделей на основе комбинированных критериев.

Последний этап построения математических моделей заключается в пересчете (адаптации) их параметров, необходимом для минимизации какого-либо (обычно, комбинированного) критерия, и позволяющим улучшить точность прогноза и добиться более полного соответствия свойств процесса и модели [14].

Адаптация может производиться для одной наилучшей модели (после селекции) и для всех структур моделей (по селекции), так что отбор производится по улучшенным значениям критерии селекции.

Среди составляющих комбинированного критерия могут быть такие, которые характеризуют только структуру модели, независимо от ее коэффициентов (например, показатели смещения). При этом адаптация сводится к минимизации подбором коэффициентов модели только второй составляющей комбинированного критерия (если она является характеристикой и структуры, и параметров), что упрощает расчеты и допускает применение градиентных процедур и методов слу- чайного поиска.

#### Краткая характеристика программ

главе описаны четыре программы, предназначенные для решения задач прогнозирования на основе выделения трендов.

В первой из них реализованы характеристики комбинаторных схем МГУА методы усложенияния моделей. В связи с этим возможна построение только сравнительно простых трендов — трендов с небольшим числом аргументов.

Вторая является типичной программой многорядных (селективных) алгоритмов МГУА и позволяет формировать сложные прогнозирующие модели с большим числом аргументов и высокими степенями по времени. В этих двух алгоритмах используются алгебраические опорные функции, и их можно применять при анализе регулярных временных рядов.

Третья программа предназначена для прогнозирования циклических временных рядов. Тренд процессы здесь представляются в виде полигармонической функции с возможностью нес端正ыми и несоответствующими к интервалу наблюдения периодами гармоник.

В четвертой программе для выделения трендов используются модели типа авторегрессии с некоторыми усложениями.

## 2. ПРОГРАММА СИНТЕЗА МОДЕЛЕЙ ДЛЯ ДОЛГОСРОЧНОГО ПРОГНОЗИРОВАНИЯ

Программа (прогр. 1) предназначена для построения полиномиальных временных трендов и используется при решении задач долгосрочного прогнозирования [13, 16, 20].

Язык программирования FORTRAN, ЭВМ БЭСМ-б Используемые подпрограммы и функции: VEER, KOMBI, PRI BCOUNT, COUNT, INITL, SHIFT, COEF, REPLACE, GAUSS, FINAL, BAL, FBAL.

Метод. Для построения прогнозирующих моделей группы взаимосвязанных переменных применяется комбинаторная схема алгоритмов самоорганизации, в результате чего формируется набор полиномиальных моделей по каждой переменной. Из полученного множества отбирается по критерию селекции пять наилучших комбинаторных моделей.

#### Описание параметров:

ITP — величина, равная единице, если строятся временные тренды;

ID — число независимых переменных (или максимальная степень полинома плюс единица для временных трендов);

NT — длина обучающей последовательности;

IH — номер точки начала прогноза; IPR — число точек, для которых вычисляются прогнозируемые значения (включая и точки расчета аппроксимации); NB — номер точки, начиная с которой вычисляется показатель баланса;

KB — номер точки, до которой включительно вычисляется показатель затягиваний баланса; LL — величина, равная нулю, если функция баланса не используется, и единице в противном случае;

KK — число функций баланса;

LK — параметр, равный единице (если критерий селекции балансовый), двум (если комбинации трендов выбираются по нес端正ности) и трем (если используется комбинированный критерий);

K — число взаимосвязанных переменных, составляющих систему (без учета функций баланса);

ICM — параметр, определяющий разницивность критерия селекции: если  $ICM = 1$ , — минимум смещения, если  $ICM = 2$ , — интегральный;

$Y(K, NT)$  — временные ряды K переменных;  $X(ID, NT + IPR)$  — массив независимых переменных (вводится только для случая  $ITP = 0$ ).

Размерности рабочих массивов задаются следующим образом:  $IS(ID)$ ,  $IK(ID)$ ,  $A(ID)$ ,  $A1(ID)$ ,  $KS(ID)$ ,  $X(ID, NT + IPR)$ ,  $Y(NT)$ ,  $C(ID, ID + 1)$ ,  $PR((K + KK) * (2^{**} ID - 1))$ ,  $IPR(K + KK)$ ,  $CM((K + KK) * (2^{**} ID - 1))$ ,  $B(K)$ ,  $BEST(K + KK + 3, 5)$ ,  $B1(K + KK, 3)$ ,  $YY(K, NT)$ ,  $Y1(K + KK, NT)$ .

Последовательность работы программы и ее основные характеристики. Прежде всего с помощью программы VEER, реализующей комбинаторный алгоритм МГУА [20], строятся таблицы постепенного усложенияния полиномов для каждого из аргументов, к которым относится заранее сформированные с помощью функции FBAL дополнительные переменные. Коэффициенты полиномов определяются по методу наименьших квадратов по всем точкам таблицы исходных данных. Кроме того, каждый полином получает свою оценку — значение смещения.

После этого в программе с помощью подпрограммы KOMBI осуществляется перебор комбинаций моделей по однотипной таблице усложения и отбираются пять лучших комбинаций по заданному критерию селекции (критерий баланса, критерий нес端正ности, комбинированный критерий). Вид критерия selection задается программистом (параметром LK).

В программе предусмотрены широкие возможности для формирования основного и вспомогательного критерия селекции. Наприимер, эффективность получающихся моделей может быть оценена следующим образом:

по показателям смещения как в обычной форме, связанной с последовательным делением выборки на две части, так и в интегральной форме, по показателям смещения различных функций баланса переменных (выбор этих функций осуществляется программистом с учетом физической сущности решаемой задачи);

по комбинированным критериям, которые могут формироваться как на основе различных критериев баланса, так и с учетом критериев нес端正ности и баланса переменных одновременно.

В связи с тем, что рассматривается система взаимосвязанных параметров (переменных), характеризующих исследуемый процесс, применяются так называемые «системные показатели нес端正ности», учитывающие смещение отдельных компонентов в следующем виде [3]:

$$n_{cm}^* = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{n_{cm_i}}{n_{cm \text{ max}}} \right)^2}, \quad (2.26)$$

где  $n$  — число параметров (переменных);  $n_{cm_i}$  — погрешность  $i$ -й модели;

$$n_{cm \max} = \max_i n_{cm_i}$$

При формировании комбинированных критерев приняты единичные веса для нормированных составляющих баланса и несмещенностии:

$$K = \sqrt{\left(\frac{B}{B_{\max}}\right)^2 + \left(\frac{n^*}{n_{cm \max}}\right)^2}, \quad (2.27)$$

где  $B$ ,  $n^*$  — небаланс и смещение для конкретной комбинации моделей системы;

$B_{\max}$ ,  $n_{cm \max}$  — их максимальные значения на множестве возможных комбинаций.

При работе программы выводятся на печать:

коэффициенты ( $ID^2 - 1$ ) моделей и их смещения для каждой исходной переменной и функции баланса, а также рассчитанные по ним прогнозируемые значения на интервале IPR;

результаты секции по выбранному критерию, сведенные в таблицу.

**Комментарий.** Для обеспечения приемлемого объема памяти и времени решения задачи прогнозирования целесообразно использовать программу при  $ID \leq 5$  и  $K + KK \leq 5$ .

#### ПРОГРАММА 1

```

1. PROGRAM FORGST
2. DIMENSION IS(4),A(4),IS(4),X(4,35),Y(10),C(4,5),PR(45,25),
   *A1(4),IT(3),CM(45),BEST(6,5),B(3),BI(1,3),YY(2,10),IK(4),
   **Y1(3,10),YY1(10,2)
4. 100 FORMAT (10F4.2)
5. 101 FORMAT (2X,'YY',10F8.2)
6. 103 FORMAT (I2)
7. 104 FORMAT (12I6)
8. 105 FORMAT (3X,IT',4XID',4XMT',4XIH',3XIPR',4XNB',
   *4X'KB',4X'LL',4X'MR',4X'LR',5X'K',3X'TSM')
11. 106 FORMAT ('TRENDS FOR VARIABLE',I3,/,1X,22(I1--))
12. 107 FORMAT ('',TRENDS FOR BALANCE VARIABLE',I3,/,1X,30(I1--))
13. 108 FORMAT ('',INPUT DATA')
14. READ 103,IT,ID,NT,IH,IPR,NB,KB,LL,MR,LR,NT,CM
15. PRINT 105
16. PRINT 104,IT,I1,NT,IH,IPR,NB,KB,LL,MR,LR,IOM
17. READ 100,YY1
18. PRINT 108
19. PRINT 101,YY1
20. DO 20 I=1,I1
     23. DO 1 I=1,IT
        24. 1 IK(I)=0
21. DO 20 J=1,NT
22. 20 YY1(J,I)=YY1(J,I1)

```

#### ПОДПРОГРАММА VEER

Построение заданного числа трендов (всего трендов) различных структур для одной переменной, вычисление показателей смещения для каждого из них, вывод на печать коэффициентов и прогнозных значений (прогр. 2).

*Обращение.* CALL VEER (X, Y, NT, NT1, ID, IH, IPR, NV, PR, K, CM,

IS, A, IK, C, AI, CM, ID1)

*Описание параметров.*

PR — выводимый массив аппроксимаций временного ряда и прогнозов, рассчитанных по трендам различной структуры. Размерность массива NV \* IPR;

X — вводимая матрица значений всех аргументов тренда на исходной выборке и на интервале расчета аппроксимации и прогнозов. Размерность матрицы ID \* NT1;

Y — вводимый вектор значений временного ряда. Длина вектора равна NT;

IK — вводимый вектор, определяющий максимальную сложность тренда. Длина вектора ID;

CM — выводимый вектор показателей смещения для каждого из трендов. Длина вектора NV;

NT — длина исходной выборки, NT1 = NT + IPR;

ID — количество аргументов тренда. Следующие IPR точек образуют значения аргументов, используемых для расчета аппроксимации и про-

```

26. NV=(1D*2-1)*(K+IK)          46.    DO 8 I=K2,K3
27. K1=ID*K2-2-1               47.    8 Y1(I,1,I)=FBAL(B,I,K,K,I)
28. IT(IT,IO,1)COTO 2          48.    9 CONTINUE
29. READ 100,X                   49.    K=K+IK
30.      GOTO 6                  50.    DO 11 J=1,J
31.      2 J1=IH-1                51.    DO 10 J=1,IT
32.      DO 51=1,NT1                52.    10 Y(J)=Y1(1,J)
33.      IF(I,GT,NT)I1=II+1       53.    11=K-I
34.      DO 5J=1,ID                54.    12=IK-II
35.      IT(I-NV)4,4,3            55.    TF (II,CE,K)PRINT 106,I
36.      3 X(J,I)=II**(J-1)       56.    TF (I2,GT,O)PRINT 107,I2
37.      GOTU 5                  57.    CALL VVER(X,Y,NT,NT1,ID,
38.      4 X(J,I)=I**(J-1)         58.    *IT,IPR,NV,PR,I,CM,IS,
39.      5 CONTINUE                59.    *A,IK,C,AL,ICM,ID+1)
40.      6 K2=II+1                 60.    11 CONTINUE
41.      K=K+IK                   61.    CALL KOMBI(NV, ID, K, I1,
42.      42.    DO 9 II=1,NT           62.    *J1,B,KB,CM,PR,IPR,BEST,
43.      43.    DO 7 J=1,I1           63.    *K+3,IK,LL,IK,IT,B,B)
44.      44.    B(J)=YY(J,I1)        64.    CALL PR(PR,BEST,K+3,IK,K)
45.      7 Y1(J,I1)=B(J)          65.    END

```

IPR — число запоминаемых значений аппроксимации исходного временного

ряда и прогнозов;

K — номер переменной, для которой строится «всев» прогнозов,

ICM — параметр, определяющий разновидность критерия минимума смеще-

ния (при  $ICM = 1$  используется критерий в обычной форме, а при

$ICM = 2$  — в интегральной);  $ID_i = iD + l$ ;

IS, A, AI — рабочие векторы длиной ID;

C — рабочая матрица размерностью  $ID * ID$ .

FINAL — требуемые подпрограммы и функции. INITL, COEF, SHIFT, BCOUNT,

Metod. Реализован комбинаторный алгоритм МГУА построения системы

прогнозов («всев» прогнозов) для некоторой переменной. Структура трендов ко-

дируется в двоичном коде в векторе — индикаторе структур IS. Для изменения

структур используется двоичный счетчик. В программе предусмотрена возмож-

ность последовательного построения систем трендов для различных переменных

с накоплением и запоминанием результатов в массивах PR и CM.

## ПРОГРАММА 2

```
1.      SUBROUTINE VERR(X,Y,NT,NT1,ID,IH,IPR,NV,
2.      *PR,K,C1,IS,A,IT,C,CM,IM)
3.      DIMENSION X(ID,NT1),Y(NT),PR(NV,IPR),IS(ID),
4.      *A(ID),C(ID,IM1),M1(ID),IK(ID),T(1,10),CM(NV)
5.      REAL R0
6.      CALL INITL(IS, ID, 1)
7.      101 FORMAT('COEFFICIENTS') 25.   3 R0=R0-X(I,J1)*M1(I)
8.      102 FORMAT(6(E14.5))   26.   4 PR(IT,J)=R0
9.      103 FORMAT(10(E11.3))  27.   PRINT 101
10.     104 FORMAT('PROGN02')  28.   PRINT 102,A
11.     105 FORMAT('SHIFT',E11.3) 29.   I=0
12.     15=IH                30.   PRINT 104
13.     N=1                  31.   5 DO 8 J2=1,10
14.     IT=(ID**2-1)*(K-1)    32.   I=I+1
15.     1 IT=IT+1             33.   IR=I-IPR
16.     M5=1                  34.   6 T(J2)=PR(IT,I)
17.     CALL COEF(X,Y,NT,0,IS, 35.   GOTO 8
18.      *ID,N,A,C,N+1,IER,0) 36.   7 T(J2)=0.0
19.     IF(IER.EQ.0)M5=0      37.   8 CONTINUE
20.     DO 4 J=1,IER          38.   PRINT 103,T
21.     J1=J+15                39.   IF(I.LT.IPR)GOTO 5
22.     R0=0.0                 40.   CALL SHIFT(CM(I)),
23.     IF(M5).4,4,2           41.   *A,A1,ID,IS,X,Y,
24.     2 DO 3 I=1,1D           42.   *NT,ICM,N,C,N+1)
```

44.	PRINT 105,CM(IT)	47.	*EQ.0)GOTO 1
45.	CALL BCOUNT(IS, ID, N)	48.	RETURN
46.	IF(FINAL(IS, ID, IK))	49.	END

## ПОДПРОГРАММА KOMBI

Выбор наилучшей комбинации трендов из системы трендов для нескольких переменных, комбинированному (прогр. 3).

Обращение. CALL KOMBI (NV, ID, K, K1, NB, KB, CM, PR, IPR, BEST,

L, KK, LL, IK, IT, B, B1)

Описание параметров:

CM — вводимый вектор показателей смещения для трендов. Первые 2ID — 1

числа характеризуют различные тренды первой переменной, такое же

число последующих — второй и т. д. Длина вектора NV.

PR — вводимая матрица прогнозов на каждой из переменных. Размерность

BEST — выбранная матрица, характеризующая оптимальный набор прогно-

зирующих трендов по всем переменным и их параметры;

KK — число переменных, образующих систему, с учетом функций баланса;

NV — общее число трендов для всех переменных. NV = K \* (2<sup>ID</sup> - 1);

K1 — число трендов по каждой из переменных, участвующих в переборе;

ID — число аргументов наиболее стоковых трендов;

IPR — суммарная длина интервалов аппроксимации и прогнозирования;

NB, KB — начальная и конечная точки расчета небаланса;

LL — параметр, равный единице, если функция баланса используется,

LK — параметр, равный единице, если селекцию необходимо провести по

критерию баланса переменных; LK = 2 — по критерию минимума се-

лекции; LK = 3 — по комбинированному критерию;

IT, B — рабочие векторы длиной K;

B1 — рабочая матрица размерностью KK \* 3.

Требуемые подпрограммы и функции. BAL, COUNT.

Metod. Поиск наилучшей системы прогнозирующих трендов для нескольких

взаимосвязанных переменных производится поочередным опрошиванием (перебо-

ром) всевозможных комбинаций моделей и оцениванием их по выбранному кrite-

рию селекции. В результате отбираются пять наилучших систем прогнозов.

## ПРОГРАММА 3

```
1.      SUBROUTINE KALBKW(ID,K,K1,NB,KB,SM,PR,IPR,
2.      *BEST,L,IK,LL,LK,IT,B,B1)
3.      DIMENSION CM(NV),PR(NV,IPR),BEST(L,5),IT(K),B(IK),B1(KE,3)
4.      DO 2 I=1,NV
5.      IF(I.GT.1)GOTO 1
6.      CM=CM1
7.      GOTO 2
8.      1 IF(CM(I).GT.CM(J))CM=CM(I)
9.      2 CONTINUE
10.     H2=ID**2-1
```

```

11.    IF(LL.EQ.0)GOTO 10
12.    DO 3 I=1,K
13.    IF(I.LE.KK) B1(I,3)=0.0
14.    C
15.    3 IT(I)=1
16.    4 DO 5 I=1,KK
17.    B1(I,2)=0.0
18.    5 B1(I,1)=0.0
19.    DO 8 J=MB,KB
20.    DO 6 I=1,K
21.    IT(I)=IT(I)+K2*(I-1)
22.    B(I)=PR(I1,J),
23.    DO 7 I=1,KK
24.    IT=K-KK+I
25.    B1(I,1)=B1(I,1)+VAL(B,I,KK,I)**2
26.    T B1(I,2)=B1(I,2)+B(I1)**2
27.    8 CONTINUE
28.    DO 9 I=1,KK
29.    B(I,1)=SORT(B1(I,1)/B1(I,2))
30.    IF(B1(I,1).GT.B1(I,3))B1(I,3)=B1(I,1)

31.    9 CONTINUE
32.    CALL COUNT(IT,K,K1,KOH)
33.    IF(KOH.EQ.0)GOTO 4
34.    GOTO 12
35.    10 DO 11 I=1,KK
36.    11 B1(I,3)=1.0
37.    12 IZ=0
38.    DO 13 I=1,K
39.    13 IT(I)=1
40.    CM=1.0
41.    B1(1,3)=1.0
42.    14 IZ=IZ+1
43.    B2=0
44.    B3=0
45.    DO 15 I=1,KK

61.    19 B1(I,1)=SORT
62.          *(B1(I,1)/B1(I,2))
63.    GOTO 22
64.    20 DO 21 J=1,KK
65.    21 B1(J,1)=0.0
66.    22 B2=0.0
67.    DO 23 J=1,K
68.    IT=IT(J)+K2*(J-1)
69.    23 B2=B2+(CM(I1)/
70.          *CM(J))*2
71.    B2=SORT(B2/(I-KK))
72.    24 B1=0.0
73.    B1=0.0

74.    IF(LL.EQ.0)GOTO 26
75.    DO 25 I=1,KK
76.    25 B1=B1+(B1(I,1)/
77.          *B1(I,3))*2
78.    BK=SORT(BK/BK)
79.    BK1=BK
80.    IF(LL.EQ.1)
81.          *BK=SORT(B2**2+BK**2)
82.    IF(LL.GT.5)GOTO 28
83.    DO 27 I=1,K
84.    27 BEST(I,I2)=TT(I)
85.    BEST(K+1,I2)=B2
86.    BEST(K+2,I2)=BK1
87.    BEST(K+3,I2)=BK
88.    GOTO 31

89.    28 IT=K+KK
90.    91.    B3=BEST(I1,1)
91.    DO 29 J=2,5
92.    93.    IF(BEST(I1,J).LE.B3)GOTO 29
93.    B3=BEST(I1,J)
94.    94.    IF(BEST(I1,J).LE.B3)GOTO 29
95.    95.    I3=J
96.    96.    29 CONTINUE
97.    97.    IF(LK.EQ.0.1)BB=B2
98.    98.    IF(LK.EQ.2.0.RL.LK.EQ.3)BB=B1
99.    99.    IF(BB.GE.BEST(I1,I3)) GOTO 31
100.   DO 30 I=1,K
101.   101.   30 BEST(I,I3)=IT(I)
102.   102.   BEST(K+1,I3)=B2
103.   103.   BEST(K+2,I3)=BK1
104.   104.   BEST(K+3,I3)=BK
105.   105.   31 CALL COUNT(IT,K,K1,KOH)
106.   106.   IF(KOH.EQ.0)GOTO 14
107.   107.   RETURN
108.   108.   END

```

## ПОДПРОГРАММА PRI

Вывод на печать информации о пяти наилучших комбинациях прогнозируемых моделей для системы взаимосвязанных переменных (прогр. 4).

*Обращение* CALL PRI(BEST, L, LK, K).

*Описание параметров*

BEST — вводимая матрица показателей, характеризующих структуру и свойства лучших комбинаций трендов развернутой системы; L = K + 3.

LK — параметр, определяющий тип критерия селекции (см. описание подпрограммы COMB1).

K — число первичных, образующих систему; L = K + 3.

*Требуемые подпрограммы и функции* Нет.

*Характер представления результатов* Пять лучшим комбинациям трендов соответствуют пять строков формируемой при выводе таблицы В. Каждом столбце указываются номера трендов для всех переменных, образующих данную систему прогнозов, а также значения смещения

```

4.      SUBROUTINE PRI(BEST, L, LK, K)
2.      DIMENSION BEST(L,5)
3.      100 FORMAT(///)
4.      101 FORMAT (' * TREND FOR VARIABLE ',I2,2I1)
5.      102 FORMAT(//,5X,'T A B L E O F R E S U L T S')
6.      103 FORMAT(46X,'FIVE BEST COMBINATIONS')
7.      104 FORMAT (24X,5(I14))
8.      105 FORMAT (1H*,24X,5(I14))
9.      106 FORMAT (1H*,24X,5(E14.4))
10.     107 FORMAT (' * CRITERION FOR VARIABLE ')
11.     108 FORMAT (' * BALANCE',15X,I11)
12.     109 FORMAT (' * COMBINED CRITERION',4X,I11)
13.     110 FORMAT (1H*,96X,'SELECTION CRITERION')
14.     111 FORMAT (3X,95(I14))
15.     PRINT 100
16.     PRINT 102
17.     PRINT 103
18.     PRINT 104, (I,I=1,5)      25.      M=M+1
19.     PRINT 111                  26.      I=M,EQ,1)PRINT 107
20.     DO 1 I=1,K                 27.      I=I,EQ,2)PRINT 108
21.     PRINT 104,I                 28.      IF(I,EO,3)PRINT 109
22.     1 PRINT 105,                 29.      PRINT 106,
23.     *(BEST(I,J),J=1,5)          30.      *(BEST(K1,J),J=1,5)
24.     I=M 2 I=1,3                 31.      IF(I,EO,LK)PRINT 110

```

## ПРОГРАММА 4

```

1.      SUBROUTINE FBAL(B, K, KK, I)
2.      DIMENSION BB(2)
3.      BB=B(2)**2+6*B(2)+6.4*B(1)-3.8
4.      100 FBAL=BB
5.      RETURN
6.      END

```

Задает преобразование системы исходных переменных для дополнения их вспомогательными переменными (прогр. 5)

*Обращение* FBAL(B, K, KK, I)

*Описание параметров*

B — входной вектор значений исходных переменных. Длина вектора K, KK — число первичных функций баланса,

I — номер используемой функции баланса. Текст функции FBAL оформляется программистом в соответствии с конкретной задачей.

## ПРОГРАММА 5

```

1.      FUNCTION FBAL(B, K, KK, I)
2.      DIMENSION BB(2)
3.      BB=B(2)**2+6*B(2)+6.4*B(1)-3.8
4.      100 FBAL=BB
5.      RETURN
6.      END

```

## ПОДПРОГРАММА SHIFT

Рассчитывает критерий несмещенности (прогр. 6)

*Обращение* CALL SHIFT(CM, A, A1, ID, IS, X, Y, NT, ICM, N, C, NI).

*Описание параметров*

CM — значение критерия несмещенности, A (ID) — вектор коэффициентов на выборке A,

A1 (ID) — вектор коэффициентов на выборке B

*Требуемые подпрограммы и функции* COEF

*Метод* Критерий несмещенности вычисляется по формуле:

$$n_{cm} = \sqrt{\frac{\sum_{i=1}^N (y_{A_i} - y_{B_i})^2}{\sum_{i=1}^N y_{tab_i}^2}},$$

где N — длина выборки,  $y_{A_i}$ ,  $y_{B_i}$  — аппроксимация табличных данных на выборках A и B,  $y_{tab_i}$  — значения выходной величины

```

1.      SUBROUTINE SHIFT(CM, A, A1, ID, IS, X, Y, NT, ICM, N, C, NI)
2.      DIMENSION CM(ID), A1(ID), IS(ID), X(ID, NT), Y(ID, NT), C(N, ID)
3.      C1=0.0
4.      C2=0.0
5.      IF(ICM,EO,2)GOTO 3

```

```

29.      PRINT 106
30.      *(BEST(K1,J),J=1,5)
31.      IF(I,EO,LK)PRINT 110

```

## ПРОГРАММА 6

```

1.      SUBROUTINE SHIFT(CM, A, A1, ID, IS, X, Y, NT, ICM, N, C, NI)
2.      DIMENSION CM(ID), A1(ID), IS(ID), X(ID, NT), Y(ID, NT), C(N, ID)
3.      C1=0.0
4.      C2=0.0
5.      IF(ICM,EO,2)GOTO 3

```

```

6.      NT1=NT/2
7.      CALL COEF(X,Y,NT,NT1,IS, ID,N,A,C,N1,IER,0)
8.      IF(IER.NE.0)GOTO 4
9.      NT1=-NT1
10.     CALL COEF(X,Y,NT,NT1,IS, ID,N,A1,C,N1,IER,0)
11.     IF(IER.NE.0)GOTO 4
12.     DO 2 I=1,NT
13.     C3=0.0
14.     DO 1 J=1,1D
15.     1 C3=C3+X(J,I)*(A(J)-A1(J))
16.     C1=C1+C3**2
17.     2 C2=C2+Y(I)**2
18.     C1=SQRT(C1/C2)
19.     GOTO 5
20.     3 CALL COEF(X,Y,NT,0,IS, ID,N,A1,C,N1,IER,1)
21.     GOTO 10
22.     4 CM=10000.0
23.     GOTO 6
24.     5 CM=C1
25.     6 RETURN
26.     END

```

### ФУНКЦИЯ FINAL

Определяет, окончен ли процесс наращивания сложности модели ( $J = 1$ ) или нет ( $J = 0$ ) (прогр. 7)

*Обращение*  $J = 1$  FINAL (IS, ID, IK)  
*Описание параметров* см подпрограмму VEER  
*Требуемые подпрограммы и функции* Нет

#### ПРОГРАММА 7

```

1.      INTEGER FUNCTION FINAL(IS, ID, K)
2.      DIMENSION IS(ID), K(ID)
3.      DO 1 I=1, ID
4.      IF(K(I).NE.K(1))GOTO 2
5.      1 CONTINUE
6.      J=1
7.      GOTO 3
8.      2 J=0
9.      3 FINAL=J
10.     RETURN
11.     END

```

```

b1 =  $\frac{\sum_{t_1}^{T_2} [\Phi - (q_1 - q_2)]^2}{\sum_{t_1}^{T_2} \Phi^2} \rightarrow \min,$ 

```

где  $[T_1, T_2]$  — интервал времени, расположенный вокруг даты прогнозирования. В качестве входных данных использовались статистические данные за 24 г. (с 1950 до 1974 г.). По входным данным при помощи метода наименьших квадратов были получены по 31 полинома для каждой из двух перенятых степеней (начиная от простейшего  $q = a_0$  до полинома третьей степени  $q = a_0 + a_1 t + a_2 t^2 + a_3 t^3$ ). Если они почти одинаково точны на интервале интерполяции. Но на интервале экстраполяции полиномы образуют так называемые «хвоста трендов». Перебором была найдена комбинация трендов, которая является наилучшей по критерию баланса.

Но результат такого перебора при некоторых условиях не является единственным. В рассматриваемой задаче около 31 комбинации трендов (из общего числа 31!) дают одно и то же значение критерия, равное нулю. Чтобы избежать неоднозначности, был применен комбинированный критерий «несмененность плюс баланс переменных»

$$K_2 = \sqrt{n_{cm}^2 + b_1^2} \rightarrow \min.$$

Минимум комбинированного критерия соответствует следующей единственной комбинации трендов:  $q_1 = 1.622t + 0.018t^2$ ,  $q_2 = 8.604t + 0.144t^2$ , где  $t$  — (текущий год 1950) и  $q_1 = q = 86,31$ .  
 По найдено алгебраической модели получаем для 2000 г. ( $t = 50$ )  $q = q_1 + 86,31 = 427,7$  тыс. жителей.

### 3. ПРОГРАММА ВЫДЕЛЕНИЯ ПРОГНОЗИРУЮЩЕГО ТRENDA ОПТИМАЛЬНОЙ СЛОЖНОСТИ ПО МНОГОРИДНУМУ АЛГОРИТМУ МГУА

Программа (прогр. 8) предназначена для построения сложных прогнозирующих моделей, которые, кроме различных степеней времени, могут включать большое число дополнительных аргументов; значения этих аргументов на интервале экстраполации предполагаются известными (так называемые нормальные прогностирующие модели) [38].

Язык программирования АЛГОЛ, ЭВМ БЭСМ-Б.

Используемые подпрограммы и функции. GAUSS (алгольный вариант), RMA, SUM.

**Метод.** В программе реализован вариант многогрядного алгоритма МГУА [30] с линейными частными описаниями, в результате чего полное описание прогнозируемого тренда представляется в виде

$$q = a_0 + a_1 x_1 + \dots + a_m x_m + a_{m+1} x_{m+1} + \dots + a_k x_k, \quad (2.28)$$

где  $x_1 = t$ ,  $x_2 = t^2$ , ...,  $x_m = t^m$ ,  $x_i = m+1, m+2, \dots, k$  представляют собой дополнительные аргументы ( $m$  — максимальная степень временной составляющей тренда — задается априори).  
 При синтезе моделей применяется критерий регуляризации, причем разделение экспериментальных точек на проверочную и обучающую последовательности задается программистом в специальной таблице. Оптимальный тренд в программе

особым образом не отмечается, поэтому его поиск осуществляется программистом при анализе результатов счета по минимуму критерия регулярности (ошибки на пропечатанной последовательности). Для упрощения программы «назадврывающиеся многорядной структуры частных описаний, образующих оптимальный тренд» (2.28) соответствует операции следует выполнить вручную или же дополнить программу процедурами по образцу прогр.

При использовании результатов необходимо иметь в виду, что исходный временный ряд  $x_1, x_{m+1}, \dots, x_k$  масштабируется по максимальным (по модулю) значениям. Пересядные  $x_2, \dots, x_m$  формируют как соответствующие степени масштабированного  $x_1$ . Таким образом, прогнозирующий тренд (2.28) представляется, фактически, в измененном масштабе, что необходимо учитывать, пользуясь полувинчентной моделью.

**Описание параметров:**

- M** — максимальная степень временного тренда;
- N** — число точек интерполяции;
- L<sub>j</sub>** — число рядков селекции;
- R** — длина обучающей последовательности;
- R<sub>i</sub>** — длина проверочной последовательности;
- M<sub>1</sub>** — число дополнительных аргументов;
- P<sub>i</sub>** — параметр, указывающий на необходимость печати промежуточных переменных. Такая печать производится только при  $P_i = 1$ ;
- Ф** — исходный временной ряд, подлежащий прогнозированию;
- TR** — массив, определяющий моменты времени наблюдения;
- T<sub>i</sub>** — массив, задающий перестановку исходных данных для последующего дальнейшем их деления на обучающую и проверочную последовательности.

Далее необходимо ввести **M** массивов, соответствующих значениям дополнительных переменных программы. После нормирования переменных и формирования степеней первого аргумента образуются возможные пары аргументов, составляющих частные модели, определяются их параметры и вычисляются значения критерия регулярности, осуществляется селекция (отбор) лучших моделей, выходы которых становятся входами моделей последующего ряда. После этого процедура повторяется.

При работе программы на печать выдаются: исходные линейные задачи (для контроля правильности подготовки информации); величины масштабирующих множителей; на каждом ряду селекции информация о структуре и параметрах лучших частных моделей. При необходимости могут быть выведены на печать значения промежуточных переменных на интервале интерполяции (в исходном масштабе).

Вся выводимая цифровая информация сопровождается текстовыми комментариями, полностью раскрывающими ее смысл.

Густь на печать выведена информацией, представленной в табл. 2. Предположим, что  $\delta_1 = \delta_{11}$ ,  $\delta_2 = \delta_{22}$ ,  $\delta_i = \delta_{ii}$  ( $\delta_i$  — минимальная ошибка  $i$ -го ряда),  $\delta_2 < \delta_1$ , и  $\delta_2 < \delta_3$ . Тогда в качестве решения выбираем лучшую модель второго

Структура и параметры лучших моделей

Ряд аргумента	Номер	Коэффициенты			Ошиб. на проверочной последовательности
		$a_{01}$	$a_{11}$	$a_{21}$	
1	1	1	2	$a_{01}$	$\delta_{11}$
	1	1	3	$a_{02}$	$\delta_{12}$
	2	3	$a_{03}$	$a_{13}$	$\delta_{13}$

Таблица 2

ряда (в данном случае вторую):

$$\hat{\Phi} = z_2 = b_{02} + b_{12}y_1 + b_{22}y_3,$$

где  $y_1 = a_{01} + a_{11}x_1 + a_{21}x_2$ ,  $y_3 = a_{03} + a_{13}x_2 + a_{23}x_3$ . Комментарий. Возможности применения алгоритма ограничиваются неравенством  $(M+M_1)*N < 5000$ .

#### ПРОГРАММА 8

```

1. 'ALGOL';'REAL';SUM,MA,GAUSS,'BEGIN';INTEGER'M,N,L1,R,R1,M,
2. S,P,I,J,T,H,F,G,,READM,N,L1,R,R1,M1,J),'PRINT',//,5X,
3. 13MAX.СТЕПЕНЬ T,13//,5X,17БОЛЬШЕЕ ЧИСЛО ТОЧЕК,13//,5X,
4. 2ЗНОУЧ.ПОСЛЕДОВАТЕЛЬНОСТЬ,13//,5X,2ЗИРОВ.ПОСЛЕДОВАТЕЛЬНОСТЬ,
5. 13//,5X,20ЧИСЛО РЯДОВ СЕЛЕКЦИИ,12//,5X,14ЧИСЛО ДЛJ. АРГ,
6. ТИМЕНТОВ,13,/','M,N,R,R1,L1,M1,M:=M+M1,'BEGIN';ARRAY';Φ,
7. TR,Φ1(1:N),DS(1:M),E1(1:M,1:3//),XR,XR1(1:M,1:N),A1,
8. Ρ(1:3//,A(1:3,1:4//),'INTEGER';ARRAY';TI(1:N),IH(1:M,
9. 1:2//),,'REAL';CΦ,D1,D,CT,,READP(TR,FI),CΦ:=RMA(I,1,N,
10. ABS(Φ(1://))),D1:=RMA(I,1,N,ABS(Φ(1://))),'PRINT',(2X,
11. 15ИХОДНЫЕ ДАННЫЕ,/4X,11Ф),',PRINT',(4X,10F11.5)',Φ",
12. 'PRINT',(/,4X,1НР)',,,'PRINT',(4X,10F11.5)',TR,'FOR';I:=1
13. 'STEP';I'UNTIL'N'DO';BEGIN';Φ1(1://):=Φ(ТИ(1://))/Φ(Φ),J:=CT:=
14. XR√1,I//';TR(ТИ(1://)/M1,'FOR';J:=2'STEP';1'UNTIL'M=M1,DX,
15. 'BEGIN';CT:=CΦ*D,,XR(Φ,J,I//):=CT'END','END',,'FOR';J:=1'STEP';1
16. 'UNTIL'M1'DO';BEGIN';IF';J=1'THEN';PRINT//(//,2X,4НД),
17. 'ОНШЕРЕМЕНЬЕ',,,READΦ),,'PRINT',(14)',J,,'PRINT',(4X,
18. 10F11.5)',,,DS(Φ//):=RMA(I,1,N,ABS(Φ(1://))),,'FOR';I:=1
19. 'STEP';I'UNTIL'N'DO';XR(/(M-1+J,1//):=Φ(ТИ(1://))/DS(Φ//)'END',,
20. 'PRINT',(//,5X,20ИХОДНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ,/2X,6ИБОЗН,,6X,
21. 5ИМЛСД,3X,11МАСИГ.КОСХ,,/4X,1Н-,10X,1Ф,8X,F9.4)',,СΦ,,'
22. 'FOR';I:=1'STEP';1'UNTIL'M'DO';BEGIN';PRINT',(3X,1Н,12)',,I,
23. 'IF';I'L'E'M-1'THEN';PRINT',(1Н+,15X,3Н*,11)',J,,'
24. 'IF';I=1'THEN';PRINT',(1Н+,23X,F9.4)',,DI'END';ELSE';PRINT',
25. '(1Н+,14X,4НШЕР,,12,4,F9.4)',,I-1#M1,DS(1-I#M1//)'END',,
26. 'PRINT',(//,2X,29ПОСЛЕДОВАТЕЛЬНОСТИ,20T3)',,
27. 'FOR';I:=R+1'STEP';1'UNTIL'N'DO';TI(1://),,'FOR';T:=1
28. 'STEP';I'UNTIL'1'DO';BEGIN';FOR';I:=1'STEP';1'UNTIL'M-1'DO',
29. 'FOR';H:=1,2,3,4'DO';(F/H):=0,,'FOR';H:=1'STEP';1'UNTIL'R
30. 'DO';FOR';H:=1,2,3,4'DO';(F/H):=0,,'FOR';H:=1'STEP';1'UNTIL'R

```

\* Для расчета прогнозируемых значений можно пользоваться и многорядной системой частных описаний.

```

31. 'DO', 'BEGIN' A1(1/):=1., A1(2/):=XP(1,H/)., A1(3/):=
    XP(J,H/)., 'FOR' F:=1,2,3 'DO' BEGIN A(F,4/):=A(F,4/)+A1(F/)*
32.     A1(F,4/), 'FOR' G:=1,2,3 'DO' A(F,G/):=A(F,G/)+A1(F/)*A1(G/)*
33.     A1(G/), 'FOR' G:=1,2,3 'DO' A(F,G/):=A(F,G/)+A1(F/)*A1(G/)
34.     'END', 'END', H., GAUSS(A,3,P,G1.), 'GOTO' MS., G1.'FOR' H:=1,2,3 'DO'
35.     P(H,H/):=0., MS.:=H:=SUM(H,R+1,1,N, (Phi(H)-P(1/)-P(2/)*XP(1,H))
36.     -P(3/)*XP(2,H/))*2.), S:=(H-2)*(1-1)+J., 'IF' S'LE'M'THEN'
37.     'BEGIN' DS(S/):=D1., IH(S,2/):=J., 'FOR' H:=1,2,3
38.     'DO' R(H/S,H/):=P(H,H/)'END', 'ELSE' BEG IN' D:='DS(1/)', F:=1.,
39.     'FOR' G:=2 'STEP' 1 'UNTIL' M'DO', IF'D_LT'D'THEN' BEGIN DS(F/):=D1.,
40.     D:='DS(G/H/), F:=G'END', 'IF' D1_LT'D'THEN' BEGIN DS(F/):=D1.,
41.     IH(F,1/):=I., IH(F,2/):=J., 'FOR' H:=1,2,3 'DO' R(H/F,H/):=.
42.     P(H,H/)'END', 'END', END J., 'END' I., 'FOR' F:=1 'STEP' 1 'UNTIL' M'DO'
43.     'BEGIN' I:=IH(F,1/)., J:=IH(F,2/)., 'FOR' H:=1 'STEP' 1 'UNTIL' N
44.     'DO' XH(F,H/):=R(H,F,1/)+XH(F,2/)*XP(1,H/)+R(H,F,3/)*
45.     XP(2,H/)'END', D:='DS(1/)', 'FOR' F:=1 'STEP' 1 'UNTIL' M'DO'
46.     'BEGIN' 'FOR' H:=1 'STEP' 1 'UNTIL' M'DO'
47.     'IF' D_LT' DS(F/), THEN' D:='DS(F/)'END', 'PRINT', //, 15X,
48.     5ИЖД Н,12/, ,15X,7(1Н-)), Т., 'PRINT', (2Х,10ИЖМЕР АРТ,
49.     ЧИМЕНТОВ 13Х,23НК 0 0 Ф И Ч И Е Н Т Ы,17Х,
50.     11НОЛ.ЛА ПР.П)'., 'FOR' F:=1 'STEP' 1 'UNTIL' M'DO', PRINT',
51.     '(Х,12,16,6Х,3Е15.7,6Х,Е11.5)', IH(F,1/), IH(F,2/), 'FOR',
52.     II:=1,2,3 'DO' R(H/F,H/), DS(F/), 'PRINT', (72Х,Е11.5,2Х,
53.     10ИЖН.ОШЕЖА)', Д., 'IF' II=1'THEN' BEGIN 'FOR' F:=1 'STEP' 1
54.     'UNTIL' M'DO', 'FOR' II:=1 'STEP' 1 'UNTIL' M'DO' R(H/F,T(H/)):=
55.     XP(H,F,H/)*СФ., 'PRINT', //, 7Х,15ИЖРОМ.ПЕРЕМЕННЫЕ)', 'FOR'
56.     F:=1 'STEP' 1 'UNTIL' M'DO', BEGIN 'PRINT', (4Х,2ИН=, 12)', F.,
57.     'PRINT', (8Х,10F11.5)', 'FOR' H:=1 'STEP' 1 'UNTIL' N'DO'
58.     M(H,F,H/), END', END'T, 'END', ENY', EOP'

```

#### 4. ПРОГРАММА ВЫДЕЛЕНИЯ ГАРМОНИЧЕСКОГО ТРЕНДА С НЕКРАТНЫМИ ЧАСТОТАМИ ДЛЯ ПРОГНОЗИРОВАНИЯ КОЛЕБАТЕЛЬНЫХ ПРОЦЕССОВ

Программа (прогр. 9) предназначена для прогнозирования природных колебательных процессов, гармонического анализа временных рядов, решения задач аппроксимации и выделения главной периодической составляющей процессов [8, 10, 11].

Язык программирования ФОРТРАН, ЭВМ — БЭСМ-6.  
Используемая подпрограмма HARMAN, WB, FR1Q, WCOEF, WB1, PPI.  
Метод. За исключением нескольких усовершенствованной программы на языке ФОРТРАН для ЭВМ типа БЭСМ-6 повторяет описанную программу на языке

АЛГОЛ для той же ЭВМ [10]. Процесс аппроксимации функции дискретного времени  $f_n$  на интервале  $[1, N]$  отрезком тригонометрического ряда с  $m$  некратными частотами  $\omega_k \neq \omega_{k_1}$  (т. е. частотами, несоотнесенными к периоду наблюдения)

$$y_n = \sum_{k=1}^m |A_k| \sin(\omega_k n) + B_k \cos(\omega_k n), \quad (2.29)$$

$$\omega_l \neq \omega_j, \quad i \neq j, \quad 0 < \omega_i < \pi, \quad i = 1, 2, \dots, m$$

делится на три этапа [7].

Первый этап — определение  $m$  коэффициентов  $a_{kp}$ ,  $p = 0, 1, \dots, m$  из условия наилучшего соблюдения соотношений

$$\sum_{n=0}^{m-1} a_p (f_{i+p} + f_{i-p}) = f_{i+m} + f_{i-m}. \quad (2.30)$$

На втором этапе из уравнения

$$a_0 + \sum_{p=1}^{m-1} a_p \cos(p\omega) = \cos(m\omega), \quad (2.31)$$

которое с помощью рекуррентного соотношения

$$\cos k\omega = 2 \cos((k-1)\omega) \cos \omega - \cos((k-2)\omega) \quad (2.32)$$

приводится к алгебраическому уравнению  $m$ -й степени относительно  $\cos(\omega_k)$ , находятся частоты  $\omega_k$ ,  $k = 1, 2, \dots, m$ .

*На третьем этапе* линейно входящие в выражение (2.29) параметры  $A_k$  и  $B_k$ ,  $k = 1, 2, \dots, m$  определяются при известных частотах методом наименьших квадратов (МНК).

Встроенные в программу усовершенствования состоят в том, что в число оптимизируемых параметров входит постоянная составляющая процесса, а также в возможность применять при определении параметров  $a_{kp}$  из условий (2.30) метод взвешенных пакетов квадратов [7]. Выбор весовых матриц обоснован в работе [11]. Кроме того, для повышения помехустойчивости процедуры первого этапа наследует суммирование временного ряда на скользящем интервале (стглаживание) [15].

В программе на первом этапе обрабатывается процесс

$$\tilde{f}_i = f_{i+2} - f_i, \quad i = 1, 2, \dots, N-2, \quad (2.33)$$

лишенный постоянной составляющей, однако имеющий тот же набор частот, что и исходный процесс  $f_i$ . Изменения, связанные с наличием  $\Delta_0$  введены в процедуру третьего этапа и в операции определения аппроксимаций остатков и прогнозируемых величин.

Соотношение (2.30) заставляет динамическую систему специальной структуры:

$$f_{i+1} = a_{m-1}f_k + a_{m-2}f_{k-1} + \dots + a_{m-2}f_{k-2m+3} +$$

$$+ a_{m-1}f_{k-2m+2} - f_{k-2m+1}.$$

Для оценивания параметров динамических моделей [7] целесообразно применять более сложные (по сравнению с методом наименьших квадратов) схемы аппроксимации уравнений динамики, с весовыми матрицами специальной структуры. При реализации такого подхода удобно использовать схему метода наименьших квадратов, применив ее к соответствующим образом преобразованным исходным временными рядам.

В настоящей программе по желанию программиста можно накапливать суммы видов:

$$\sum_{l=m+1}^k (f_{i+p} + f_{i-p}) \quad (2.34)$$

— для всех  $p = 0, 1, \dots, m$ ;

$$\sum_{i=0}^k (f_{i+p} + f_{i-p}) \quad (2.35)$$

— для всех  $p = 0, 1, \dots, m$  и всех  $r = m+1, m+2, \dots, k$  (в обоих случаях  $k = m+1, m+2, \dots, N-m$ ).

Эти преобразования увеличивают точность определения параметров  $\alpha_p$  из условий аппроксимации соотношений (2.30).

При сильно зашумленных данных для повышения устойчивости работы программы введен фильтрующее преобразование

$$\hat{f}_k = \sum_{i=k}^{k+T} \tilde{f}_i, \quad k = 1, 2, \dots, N-T, \quad (2.36)$$

которое не изменяет спектральный состав процесса [11]. Преобразование (2.36) можно применять при любой из реализованных схем оценивания параметров.

Положительно сказывается на точности вычислений применение специальной процедуры для нахождения только действительных корней полинома, не допускающей появления минной части [11] (в программе [10] использовалась общая процедура поиска комплексных корней). Этим снимается вычислительные погрешности решения уравнения (2.31).

**Описание параметров:**

$N$  — длина обучающей последовательности;

$NE$  — длина экзаменационной последовательности;

$PT$  — число точек прогноза;

$F$  — символы выбора;

$FM$  — максимальное число выделяемых гармоник;

$NPM$  — максимальное число рядов селекции;

$KA$  — способ аппроксимации (см. описание подпрограммы WB на с. 44);

$KT$  — читывал суммирования в фильтрующем преобразовании;

$IP$  — тип критерия селекции;

$G (N + NP + NE)$  — исходный временной ряд, подлежащий аппроксимации.

Размерности рабочих массивов задаются так, как указано в подпрограмме HARMAN (см. с. 38).

**Последовательность работы программы и ее основные характеристики.** Установление гармонических трендов является характерным для программ МГУА с последовательным выделением трендов [10]. Прежде всего из исходного временного ряда выделяются тренды с 1-й, 2-й, ...,  $m$ -м гармоническими составляющими, среди которых по определенному критерию селекции, вычисляемому на точках проверочной последовательности, находятся  $F$  (символы выбора) лучших. Для каждого из  $F$  остатков (разность между исходным времененным рядом и аппроксимацией) формируются тренды различной сложности, и из общего числа  $Fm_{\max}$  трендов второго ряда обрабатываются с шагом  $F$  наименьших. На последующих шагах селекции эта процедура повторяется. Оптимальный гармонический тренд процесса определяется по минимуму выбранного критерия селекции.

Возможности программы ограничены:

$$F \leq 15, \quad FM \leq 15, \quad (F \leq FM). \quad (2.37)$$

Все необходимые операции (выделение и отбор трендов, формирование остатков, поиск наилучшего тренда, расчет и вывод на печать его параметров и соответствующих прогнозируемых значений) выполняются в подпрограмме HARMAN, которая обращается к следующим подпрограммам:

WB — находит балансовые коэффициенты  $\alpha_p$ ,  $p = 0, 1, \dots, m-1$  для процесса

$\tilde{f}_k$  или  $\hat{f}_k$  из уравнения (2.36);

FR1Q определяет косинусы частот из уравнения (2.31), стоящие его при помощи соотношения (2.32) к полному степени  $m$  относительно  $\cos \omega_i$ ;

WB1 по известным косинусам частот  $\cos \omega_i$ ,  $i = 1, 2, \dots, m$  из уравнения (1.33) вычисляет балансовые коэффициенты  $\alpha_p$ ; WCOEF по известным частотам  $\omega_i$  определяет коэффициенты  $A_0$ ,  $A_1$ ,  $B$ ;  $j = 1, 2, \dots, m$  в разложении (2.29), наилучшим образом (по СКО) аппроксимирующем исходный временной ряд; RANG \* выстраивает элементы массива в порядке возрастания их величин. Комментарий. Для контроля всех подпрограмм следует воспользоваться программой H<sub>L</sub> TEST. При заданых в подпрограмме FR1Q величинах EPS1, EPS2, EPS3, ETA параметры полигармонической функции восстанавливаются с точностью около 0,1%. Время счета тестового примера 1 мин

#### ПРОГРАММА 9

```

1. PROGRAM H TEST
2. DIMENSION C(58),PA(4,58),RA(4,58),APR(68),
   *YB(68),TIN(4,8,1),IST(1),C(10,10)
3. INTEGER PT
4. PT=10
5. DO 1 I=1,58
   1 G(I)=1.2+3*SIN(I*.34)-2.7*COS(I*.59)+1.1*COS(I*.43)
7. *4,3*SIN(I*1.56)+2*COS(I*1.56)
8. PT=10
9. CALL HARMAN(50,4,4,58,PT,4,4,1,1,0,5,10,68,
10. *G,YB,AP,AR,PB,PA,TIN,IST,C)
11. END

```

#### ПОДПРОГРАММА HARMAN

Выделение гармонического тренда оптимальной сложности для временного ряда, представляющего собой когебатильный процесс, на основе многорядной процедуры саморганизации (прогр. 10).

Обращение. CALL HARMAN ( $N$ ,  $NP$ ,  $NE$ ,  $NN$ ,  $PT$ ,  $F$ ,  $FM$ ,  $NRM$ ,  $KA$ ,  $KT$ ,  $IP$ ,  $F2$ ,  $NPT$ ,  $G$ ,  $YB$ ,  $AP$ ,  $AR$ ,  $PB$ ,  $PA$ ,  $TIN$ ,  $IST$ ,  $C$ ).

#### Описание параметров:

$G$  — вводимый вектор, представляющий исходный временной ряд. Длина вектора  $NN$ ;

$APR$  — выводимый вектор аппроксимации и прогнозов. Длина вектора  $NPT$ ;

$N$  — длина обучающей последовательности;

$NP$  — длина проверочной последовательности;

$NE$  — длина экзаменационной последовательности (может быть равной нулю);

$NN$  — общая длина временного ряда ( $NN = N + NP + NE$ );

$PT$  — способ выбора прогноза;  $NPT = NN + PT$ ;

$F$  — способ выбора для трендов;

$FM$  — максимальное число выделяемых частот;

$NRM$  — максимальное допустимое число рядов селекции;

$KA$  — параметр, определяющий тип аппроксимации (см. описание подпрограммы WB);

$KT$  — интервал стаживания в фильтрующем преобразовании (2.36);

$IP$  — параметр, определяющий тип применяемого критерия селекции (см. описание подпрограммы PPI);  $F2 = 2 * FM + 2$ ;

\* Описание подпрограммы, отмеченный звездочкой, см. в приложении

УВ — рабочий вектор длиной NPT;

РА, РА1 — рабочие матрицы размерностью F \* NN;

TIN — рабочая матрица для представления информации о структуре оптимального тренда. Размерность матрицы F \* 8 \* NRM;

IST — рабочий вектор длиной NRM;

C — рабочая матрица размерности F2 \* F2.

Векторы F, FM, PT, F2 — типа INTEGER.

*Продолжение подпрограммы и функции:* WB, FRIO, WBL, WCOEF, PPI, PANG.

При работе подпрограммы на печать выдаются: расшифровка и значение входных переменных; исходный временной ряд; информация о характеристиках лучших трендов на каждом из рядов селекции; структура и параметры оптимального тренда; аппроксимации исходной последовательности и следующих РТ тактов времени.

*Метод.* Выполнение подпрограммы заключается: в последовательности обращений к подпрограммам WB, FRIO, WBL, WCOEF для формирования параметров частных трендов; в вычислении аппроксимаций и характеристик трендов; в их селекции и запоминании структур лучших трендов. Для сокращения объема машинной памяти расчеты аппроксимаций и тренда оптимальной сложности проводятся в копии выполнения подпрограммы повторным восстановлением параметров тренда по его структуре в процессе последовательного выделения трендов уже известной сложности.

```
      PROGRAMMA 10
1.  SUBROUTINE NARMAN(M,IP,NE,NM,PT,F,FM,NRM,KA,KT,IP,
2.  *I12,IRT,G,UV,AR,APR,RA1,TIN,IST,C)
3.  REAL TB(6)
4.  INTEGER F,FM,PT,F2
5.  DIMENSION UV(MT),PA(F,MN),PA1(F,NN),AP(IRT),APR(IRPT)
6.  IST(NRM),C(F2,F2)
7.  DIMENSION A(16),W(16),MK(16),B(17)
8.  100 FORMAT (//)
9.  101 FORMAT (5X,'СВОБОДНЫЕ ЧЛЕНЫ',I3.5)
10. 102 FORMAT (5X,'МАК.ЧИСЛО ЧАСТОТ',I3)
11. 103 FORMAT (5X,'МАК.ЧИСЛО ИДОВ',I3)
12. 105 FORMAT (5X,'ДЛИНА ПРОВ.ПОСЛЕДОВАТЕЛЬНОСТИ',I4)
13. 104 FORMAT (5X,'ДЛИНА УКАЗ.ПОСЛЕДОВАТЕЛЬНОСТИ',I4)
14. 106 FORMAT (5X,'ДЛИНА ОБЩ.ПОСЛЕДОВАТЕЛЬНОСТИ',I4)
15. 107 FORMAT (5X,'ЧИСЛО ТОЧЕК ПРОТЮЗА',I3)
16. 108 FORMAT (5X,'ЛИНГР.ПРИ МА=',I2)
17. 109 FORMAT (//)
18. 110 FORMAT (10I12.4)
19. 111 FORMAT (2X,'ВРЕМЕННОЙ РЯД')
20. 112 FORMAT (20X,'ОПТИМАЛЬНЫЙ ТРЕНД',/,20X,I3(I4))
21. 113 FORMAT (/,5X,'НОМЕР РИДА',I3)
22. 114 FORMAT (3X,'ЧИСЛО ЧАСТОТ',I3)
```

```
23. 115 FORMAT (3X,'СВОБОДНЫЕ ЧЛЕНЫ',I3.5)
24. 116 FORMAT ('ЧАСТОТЫ',9X,'A',17X,'B',16X,'АМПЛИТУДА')
25. 117 FORMAT (F10.7,3F17.6)
26. 118 FORMAT (2X,'ДЛЯ СТВ.ЗНАЧЕНИЙ')
27. 119 FORMAT (5I18.6)
28. 120 FORMAT (2X,'АППРОКСИМАЦИИ')
29. 121 FORMAT (25X,'ПРОГНОЗ',/)
30. 122 FORMAT (18,2F8.5)
31. 123 FORMAT (18,F36.5)
32. 124 FORMAT (/,2X,'В УВ')
33. 125 FORMAT (/,2X,'В WCOEF')
34. 126 FORMAT (/,2X,'В WBL')
35. 127 FORMAT (1H*,1LX,'НЕТ РЕШЕНИЙ')
36. 128 FORMAT (5X,'ИНТЕРВАЛ СГЛАЖИВАНИЯ',I2)
37. I=K+1
38. I8=0
39. I8=I+HE
40. M1=I+I8
41. MK=I8+I
42. PI=ASIN(1.0)
43. PRINT 100
44. PRINT 106,M
45. PRINT 105,IR
46. PRINT 105,NE
47. PRINT 102,FM
48. PRINT 101,F
49. PRINT 107,PT
50. PRINT 103,NRM
51. PRINT 108,MA
52. PRINT 128,KT
53. PRINT 109
54. PRINT 111
55. PRINT 110,G
56. IR=1
57. 1 IT=0
58. 2 IT=IT+1
      73. 7 W(J)=-IST(W(J))+PI
      74. CALL WBL(W,M,M+1,A,IER,C)
      75. IF (IB.EQ.0) GO TO 201
      76. CALL RANG(M,V)
      77. 201 CALL WCOEF(W,B,M,2*M+1,2*M+2,
```

```

79.    IF (I.EQ.0) GOTO 997
80.    IF (I8.EQ.1) GOTO 202
81.    B1=0.0
82.    RZ=0.0
83.    B3=0.0
84.    B4=0.0
85.    D2=0.0
86.    D3=0.0
87.    M1=M+1
88.    I1=I-1
89.    DO 11 I=ML,MIA
90.    R=0.0
91.    DO 12 J=1,M
92.    IJ=I-J-1
93.    IZ=I-J+1
94.    L2=R*I*(J)*YB(I1)+YB(I2)-2*BMP+1)
95.    IM=I+M
96.    M1=I-M
97.    R=(YB(IM)+YB(M1)-R-2*BMP+1)**2
98.    IF (I.GT.(I-M)) GOTO 13
99.    B1=B1+R
100.   GOTO 11
101.   13 IF (I.GT.(M-I)) GOTO 14
102.   B2=B2+R
103.   GOTO 11
104.   14 B3=B3+R
105.   11 CONTINUE
106.   15 B1=SQR(B1/(I-M))
107.   16 B2=SQR(B2/RB)
108.   DO 15 I=1,MR
109.   R=0.0
110.   DO 16 J=1,M
111.   D=P(I,J)*I
112.   17 R=R+B(2*I-1)*SIN(C)+B(2*I)*COS(D)
113.   D=YB(I)+YB(I-1)-B(M2+1)-R)**2
114.   18 B(M1)*R
115.   DO 17 I=1,MR
116.   19 I1=I-1
117.   R=A(I,I)
118.   DO 18 J=1,M
119.   IJ=I-J-1
120.   20 R=R+(IJ)*(A(I,J)*AP(I2))
121.   IZ=I+M
122.   A(I,I)=R
123.   D=(YB(IZ)-R-B(M+1))*2
124.   IF (IZ.GT.M) GOTO 19
125.   D=0.0
126.   GOTO 17
127.   21 IF (IZ.GT.M1) GOTO 20
128.   IZ=IZ+1
129.   GOTO 17
130.   20 D3=IZ*D
131.   17 CONTINUE
132.   22 IB(4)=SORT(D1/I)
133.   23 IB(5)=SORT(D2/IR)
134.   IF (NE) 24,21,22
135.   24 IB(3)=0.0
136.   IB(6)=SORT(D3/IE)
137.   GOTO 23
138.   22 IB(3)=SORT(B3/NE)
139.   IB(6)=SORT(D3/IE)
140.   23 IF ((IT.EQ.1).AND.(M.LE.F)) GOTO 24
141.   GOTO 25
142.   24 IF (DIP-1) 26,26,27
143.   26 TIN(W,1,NR)=0.0
144.   GOTO 28
145.   27 TIN(M,1,NR)=IT
146.   28 TIN(M,2,NR)=M
147.   DO 29 I=1,6
148.   29 TIN(U,I+2,NR)=IB(I)
149.   DO 30 I=1,NK
150.   30 TIN(W,I+2,NR)=AR(I)-BMP+1
151.   GOTO 34
152.   25 R=0.0
153.   IZ=0
154.   DO 31 I=1,F
155.   D=TIN(I,1P+2,NR)
156.   IF (R.GT.D) GOTO 31
157.   R=D
158.   1Z=F
159.   31 CONTINUE
160.   1F (R.LE.IB.FP)) GOTO 34
161.   DO 32 I=1,NK
162.   32 PA((IZ,I)*B(I)-AR(I)-BAMP+1)
163.   DO 33 I=1,F

```

```

187.      MR=I
188.      I1=J
189.      I2=J
190.      36 CONTINUE
191.      1ST(IK)=TIN(IZ,2,MR)
192.      I1=MR-1
193.      IF ((I1,IK,0) GOTO 382
194.      DO 37 I=1,IK
201.      39 Y(I)=G(I)
202.      GOTO 33
203.      40 Y(I)=0.0
204.      38 CONTINUE
205.      381 IZ=1
206.      41 M=IST(IZ)
207.      MP=-*M
208.      GOTO 200
209.      202 PRINT 113,I2
210.      PRINT 115,B(MP+1)
211.      PRINT 114,M
212.      PRINT 116
213.      DO 46 I=1,M
214.      PR=SQR(T(X(2*I-1)**2+B(2*I-1)**2))
215.      46 PRINT 117,M(I),B(2*I-1),B(2*I),P1
216.      DO 47 I=1,MP
217.      K=0.0
218.      DO 48 J=1,M
219.      D=Y(J)*I
220.      R=R+B(2*I-1)*SIN(D)+B(2*I)*COS(D)
221.      APR(I)=APR(I)+R*B(MP+1)
222.      47 AP(I)=R
223.      M1=M+1
224.      MR=MR+PT-M
225.      DO 53 I=M,NM
226.      I1=I+M
227.      R=AP(I1)
228.      DO 49 J=1,M
229.      IJ1=I+J-1
230.      IJ2=I-J+1
231.      49 R=I+A(J)*(AP(IJ1)+AP(IJ2))
232.      IZ=I+M
233.      AP(IZ)=R
234.      53 APR(IZ)=APR(IZ)+AP(IZ)+B(MP+1)

```

### ПОДПРОГРАММА WB

Вычисление балансовых коэффициентов для исходного временного ряда, представляющего колебательный процесс (прогр. 11).

*Обращение.* CALL WB (X, Y, N1, Y1, Y2, A, M11, C, IER, KA, KT).

*Описание параметров:*

X — вводимый вектор, представляющий анализируемый временной ряд.

A — выводимый вектор балансовых коэффициентов. Длина вектора M;

M — число выделяемых гармоник (число балансовых коэффициентов); KA — параметр, определяющий метод аппроксимации. При KA = 0 используются обычный метод наименьших квадратов, при KA = 1 — переменные преобразуются по формуле (2.34), а при KA = 2 — по формуле (2.35);

KT — параметр, задающий длину интервала стяживания в преобразовании (2.36);

IER — параметр, принимающий значение 0, если система нормальных уравнений не была решена, и 1 — в противном случае;

Y1, Y2 — рабочий вектор длиной M1; M11 = M + 1;

C — рабочая матрица размерностью M \* M11.

*Процедура подпрограммы и функции.* CAUSS.

*Метод.* Исторический временной ряд преобразуется в соответствии с выражением (2.33). Затем составляется и решается система нормальных уравнений, отвечающая заданному методу аппроксимации соотношении (2.30).

**ПРОГРАММА 11**

```

1. SUBROUTINE VBR(X,Y,M1,Y1,Y2,A,M,M11,C,IER,K1,KT)
2. DIMENSION X(K1),Y(K1),Y1(K1),Y2(M11),A(K1),C(M11)
3. K=K1-2
4. K1=K+1
5. M=M-KT
6. DO 1 I=1,N
7.   1 Y(I)=X(I+2)-Y(I)
8.   2 J=1,I1
9.   Y(J)=0.0
10.  DO 2 I=1,N!
11.  2 C(I,J)=0.0
12.  IF (KT.EQ.0) GOTO 30
13.  IF (K.EQ.0) GOTO 3
14.  DO 20 I=1,N
15.  B=Y(I)
16.  DO 21 J=1,KT
17.  21 B=B-Y(I+J)
18.  20 Y(I)=B
19.  DO 30 I=1,M1
20.  K=I-1
21.  R=K
22.  L=1.0/R
23.  DO 40 J=1,M1
24.  T1=I-J-1
25.  T2=I-J+1
26.  Y2(J)=Y1(J)*Y(I)+Y(I2)
27.  IF (R.L.EQ.0) Y2(J)=Y1(J)
28.  4 Y1(J)=Y2(J)
29.  8 DO 5 K1=1,M
30.  DO 5 J=K1,M1
      
```

**ПОДПРОГРАММА FR1Q**

Решение частотного уравнения относительно косинусов частот (прогр. 12).

*Обращение.* CALL FR1Q (A, M, WK, M1, FI, CO)

*Описание параметров:*

A — выводимый вектор балансовых коэффициентов. Длина вектора M;  
M — число выделяемых частот; M1 = M + 1;  
CO — рабочий вектор длиной M1;

**FI** — рабочая матрица размерностью  $M * M_1$ .

*Преобразование подпрограммы и функции ZEROS.*

*Метод.* Применение рекуррентного соотношения (2.32) частотное уравнение приводится к полиному степени M относительно косинусов искомых частот, а затем находятся его корни. Точность вычислений корней зависит от ряда внутренних параметров процедуры, значения которых при необходимости следует изменить (см. описание подпрограммы ZEROS в приложении)

**ПРОГРАММА 12**

```

1. SUBROUTINE FR1Q(A,M,WK,M1,FI,CO)
2. DIMENSION A(M1),WK(M),C(M1),FI(M1)
3. DO 1 I=1,M
4. 1 DO 1 J=1,M1
5. 1 FI(I,J)=0.0
6. 1 FI(I,I)=1.0
7. 1 IF (M.EQ.1) GOTO 27
8. 1 IF (I,2)=1.0
9. 1 IF (I,3)=2.0
10. 1 IF (M.LE.2) GOTO 2
11. 1 DO 3 I=2,M
12. 1 II=I+1
13. 1 DO 4 J=2,I1
14. 1 4 FI(I,J)=2*FI(I-1,J-1)-FI(I-2,J)
15. 1 3 FI(I,1)=FI(I-1)
16. 1 2 M2=M-1
17. 1 1 DO 5 I=1,M2
18. 1 1 DO 5 J=1,M1
19. 1 1 5 FI(M,J)=FI(M,J)-FI(I,J)*A(I+1)
20. 1 1 27 FI(M,1)=FI(M,1)-A1
21. 1 1 21 DO 6 I=1,M1
22. 1 1 6 C(I,I)=FI(N,I)
23. 1 1 EPSA=.000001
24. 1 1 EPS1=.00001
25. 1 1 EPS2=.00000001
26. 1 1 EPS3=.000001
27. 1 1 DO 66 I=1,M
28. 1 1 66 WK(I)=0.0
29. 1 1 CALL ZEROS(M1,CO,M,WK,1000,EPS1,EPS2,EPS3,EPA)
30. 1 1 RETURN
31. 1 END
      
```

## ПОДПРОГРАММА WCOEF

Вычисление коэффициентов  $A_0, A_j, B_j, j = 1, 2, \dots, M$  в разложении (2.29), аппроксимирующим исходный временной ряд по минимуму СКО (прогр. 13).

*Обращение. CALL WCOFF (W, B, M, M21, M22, N, Y, Y1, HM, IER).*

*Описание параметров:*

$\mathbf{Y}$  — вводимый вектор, представляющий анализируемый временной ряд. Длина

вектора  $N$ ;

$\mathbf{W}$  — вводимый вектор частот. Длина вектора  $M$ ;

$\mathbf{B}$  — выводимый вектор коэффициентов разложения. Длина вектора  $M21$ ;

$\mathbf{M}$  — число членов в полигаммическом разложении;

$\mathbf{IER}$  — параметр, принимающий значение 1, если коэффициенты были определены,

и 0 — при линейной зависимости системы нормальных уравнений;

$\mathbf{Y1}$  — рабочий вектор длиной  $N$ ;

$\mathbf{HM}$  — рабочая матрица размерностью до  $M21 * M22$ .

*Требование подпрограммы и функции: GAUSS.*

*Метод.* При формировании матрицы нормальных уравнений для аппроксимации исходного временного ряда полигаммической функцией (2.29) при известных частотах используются специальные приемы, минимизирующие число обращений к стандартным функциям SIN и COS. Для этого от элементов общего вида матрицы нормальных уравнений  $\sum_{i=1}^N \cos(i\omega_j) \cos(i\omega_k)$ ,

$$\sum_{i=1}^N \cos(i\omega_j) \sin(i\omega_k), \quad \sum_{i=1}^N \sin(i\omega_j) \sin(i\omega_k), \quad \sum_{i=1}^N \cos^2(i\omega_j), \quad \sum_{i=1}^N \sin^2(i\omega_j)$$

ходят к суммам  $\sum_{i=1}^N \sin(i\bar{\omega}_j)$  и  $\sum_{i=1}^N \cos(i\bar{\omega}_j)$ , для которых известны аналитические выражения как функции  $\bar{\omega}_j$  и  $N$ . Подробнее метод описан в работе [9]. В массиве  $\mathbf{B}$  коэффициенты расположены в таком порядке:  $A_1, B_1, A_2, B_2, \dots, B_M, A_0$ .

### ПРОГРАММА 13

```

1.      SUBROUTINE WCOFF(W,M,M21,M22,N,Y,Y1,HM,IER)
2.      DIMENSION W(M),B(M21),Y(N),Y1(N),HM(M21,M22)
3.      K=2*M
4.      K1=K+1
5.      D0 1 I=1,K1
6.      HM1=0.0
7.      IF (I-K) 2,2,3
8.      2  I1=INT((I+1.25)/2.0)
9.      9.  TI=(I+.1)/2.0-INT((I+.1)/2.0)
10.     DO 4 J=1,K
11.        JJ=INT((J+1.25)/2.0)
12.        TI=(J+.1)/2.0-INT((J+.1)/2.0)
13.        W1=W(I1)-W(JJ)
14.        W2=W(I1)+W(JJ)
15.        I1=I1-JJ
16.      5  IF (ABS(TI-TJ).LT.0.01) GOTO 8
17.      S1=0.0
18.      GOTO 9
19.      8  S1=N
20.      GOTO 9
21.      6  S1=SIN(N*I1/2.0)/SIN(W1/2.0)
22.      9  S2=SIN(N*I2/2.0)/SIN(W2/2.0)
23.      CN1=COS((I+1)*I1/2.0)
24.      CN2=COS((I+1)*I2/2.0)
25.      SH1=SIN((N+1)*W1/2.0)
26.      SN2=SIN((I+1)*W2/2.0)
27.      IF (TI-.25) 14,10,10
28.      10  IF (TI-.25) 13,12,12
29.      12  HM(I,J)=S1*CN1-S2*CN2
30.      GOTO 40
31.      13  HM(I,J)=S2*SN2+S1*SH1
32.      GOTO 40
33.      11  IF (TI-.25) 15,14,14
34.      14  HM(I,J)=S2*SN2-S1*SH1
35.      GOTO 40
36.      15  HM(I,J)=S1*CN1+S2*CN2
37.      40  HM(I,J)=HM(I,J)/2
38.      4  CONTINUE
39.      IF (TI-.25) 17,16,16
40.      16  Y1(1)=SIN(W(I1))
41.      41.  Y1(2)=SIN(2*W(I1))
42.      GOTO 18
43.      17  Y1(1)=COS(W(I1))
44.      44.  Y1(2)=COS(2*W(I1))
45.      45.  18  W1=COS(W(I1))
46.      46.  D0 19 J=3,N
47.      47.  Y1(J)=2.0*W1*Y1(J-1)-Y1(J-2)
48.      48.  19  HM1=HM1+Y1(J)*Y(J)
49.      49.  HM(I,J)=HM1+Y1(J-1)*Y1(J-2)*Y(J)
50.      50.  IF (TI-.25) 24,20,20
51.      51.  20  HM=SIN((I+1)*W(I1)/2.0)
52.      52.  GOTO 22
53.      24  H1=COS((I+1)*W(I1)/2.0)
54.      54.  22  HM(I,K1)=H1*SIN(K*W(I1)/2.0)/SIN(W(I1)/2.0)
55.      55.  GOTO 24
56.      3  HM(I,K1)=H1
57.      57.  H1=0.0
58.      58.  D0 23 J=1,I
59.      23  HM=H1+Y(J)
60.      60.  HM(I,K+1)=H1
61.      61.  24  IF (I-2) 1,25,25
62.      62.  25  I1=I-1
63.      63.  D0 26 J=1,I1
64.      26  HM(I,J)=HM(J,I1)

```

65. 1 CONTINUE  
 CALL GAUSS(M, K, K+1, B, IER)  
 RETURN

66. END

#### ПОДПРОГРАММА WB1

Вычисление балансовых коэффициентов при известных частотах из системы частотных уравнений (прогр. 14).  
*Обращение.* CALL WB1 (W, M, M1, A, IFR, C).  
*Описание параметров:*  
 W — выходный вектор частот. Длина вектора M;  
 A — выходимый вектор балансовых коэффициентов. Длина вектора M;  
 M — количество выделяемых частот;  
 IER — параметр, равный 0, если система частотных уравнений линейно зависима, и 1 — в противном случае; M1 = M + 1;  
 C — рабочая матрица, размерностью M \* M1.  
*Преобразование подпрограммы и функции GAUSS.*  
*Метод.* По известным частотам заполняются строки матрицы C. В каждой  $i$ -й строке — косинусы видов  $\cos p_i$ , при  $p_i = 0, 1, \dots, M$ . При решении этой системы относительные  $a_{ij}$  получаются уточненные значения балансовых коэффициентов. Подпрограмма необходима для исключения влияния на последующие расчеты погрешностей в вычислении корней частотных уравнений.

#### ПРОГРАММА 14

1. SUBROUTINE WB1(W,M,M1,A,IER,C)
2. DIMENSION W(M),A(M),C(M,M1)
3. DO 1 I=1,M
4. DO 1 J=1,M1
5. 1 C(I,J)=COS(W(I)\*(J-1))
6. CALL GAUSS(C,M,M1,A,IER)
7. RETURN
8. END

#### ПОДПРОГРАММА PPI

Табулирование показателей, характеризующих процесс селекции и качество трендов на каждом ряду селекции (прогр. 15).  
*Обращение.* CALL PPI (NR, IP, TIN, F, NRM).  
*Описание параметров:*

TIN — вводимая трехмерная матрица информации, размерностью  $F * 8 * NRM$ ;  
 NR — текущий номер ряда;  
 F — свобода выбора (число отбираемых лучших трендов);  
 NRM — максимальное возможное число рядов;  
 IP — параметр, определяющий тип критерия селекции (при  $IP = 2$  селекция по критерию баланса, а при  $IP = 5$  — по критерию регулярности).  
*Метод.* Находится тренд с наилучшим значением заданного критерия и выводится на печать, результаты считаются в виде таблицы, содержащей восемь граф. В первом графе указывается номер остатка, из которого выделяется тренд, во втором — число выделенных гармонических составляющих, в третьем — пятом — показатели небаланса уравнений (2,30), в шестой — восьмой — ошибки аппрокси-

мации, соответственно, на обучающей, проверочной и экзаменационной последовательностях. Такая информация выводится для всех F отобранных трендов. Лучший по заданному критерию тренд на каждом из рядов подчеркивается.

#### ПРОГРАММА 15

1. SUBROUTINE PPI(NR,IP,TIN,F,NRM) ПРОГРАММА 15
2. INTEGER F
3. DIMENSION TIN(F,8,NRM)
4. 10 FORMAT (//, ' РНД ', I2)
5. 11 FORMAT (2X,'РЕНД И', ' ЧИСЛО ЧАСТОТ ', 'ЗХ,', \*'БАЛАНС ОВ', 6X, 'БАЛАНС ПР', 6X, 'БАЛАНС ЭК', \*'4Х', 'ОШИБКА ОВ', 6X, 'ОШИБКА ПР', 6X, 'ОШИБКА ЭК', /)
6. 9. 13 FORMAT (58(2I1 -))
7. 14 FORMAT (//)
8. 12 FORMAT (16,116,5X,6E15.4)
10. 14 FORMAT (//)
11. K=1
12. 13 FORMAT (58(2I1 -))
13. IF (F.EQ.1) GOTO 4
14. DO 1 I=2,F
15. IF (TIN(1,1,I+2,NR)-P) 2,1,1
16. 2 P=TIN(I,1,P+2,NR)
17. I=I
18. 1 CONTINUE
19. 4 PRINT 10,NR
20. PRINT 11
21. DO 3 I=1,F
22. PRINT 12,TIN(I,1,NR),TIN(I,2,NR),(TIN(I,J,NR),J=3,8)
23. IF (I.EQ.1) PRINT 13
24. 3 CONTINUE
25. PRINT 14
26. RETURN
27. END

#### 5. ПРОГРАММА САМООРГАНИЗАЦИИ ПРОГНОЗИРУЮЩИХ МОДЕЛЕЙ ЦИКЛИЧЕСКИХ ПРОЦЕССОВ

Программа [35] (прогр. 16) предназначена для синтеза прогнозирующих моделей процесса изменения среднемесячного стока реки по наборным данным. Она позволяет также моделировать и другие циклические процессы при дискретизации периода на заданное число интервалов. Цикл программирования АЛГОЛ, ЭВМ БЭСМ-6.

**Используемые подпрограммы.** DES, MAX, NOS, PAR, GAUSS.

**Метод.** Процесс изложения во времени количества воды, протекающей в конкретном году заданном створе реки, характеризуется ярко выраженной цикличностью, и длина этого цикла равна году.

Программа синтезирует модель перераспределения годового стока реки по месяцам года в заданном створе, отражающую основные закономерности естественного процесса изменения среднемесячного стока  $\bar{P}$ , т. е. модель, позволяющую по заданному значению прогноза среднегодового стока рассчитать с достаточной точностью прогноз среднемесячных значений (расходов) стока  $P_k$ ,  $k = 1, 2, \dots, 12$ .

Такая модель называется «прогнозирующей моделью среднемесячного стока», или просто «моделью стока».

Предполагается, что среднегодовой сток прогнозируется независимо, например, по одной из программ самоорганизации.

Среднегодовые и среднемесечные расходы стока связаны соотношением

$$\bar{P} = \frac{1}{12} \sum_{k=1}^{12} P_k, \quad (2.38)$$

выражающим баланс месячных и годовых количеств воды, протекающей в заданном створе реки. Построенная по натуральным данным модель стока будет удовлетворять соотношению (2.38) с точностью до величин разбогатится или певзки:

$$b_i = \bar{P}_i - \frac{1}{12} \sum_{k=1}^{12} P_{ik}, \quad (2.39)$$

где  $i$  — индекс года.

В работе предполагается метод синтеза прогнозирующих моделей речного стока по данным наблюдений за стоком в течение нескольких лет по критерию минимума величины

$$B = \frac{1}{T} \sum_{i=1}^T b_i^2 \rightarrow \min, \quad (2.40)$$

где  $T$  — число лет наблюдений. Критерий (2.40) является, фактически, критерием баланса.

Для отыскания закономерности перераспределения среднегодового стока реки по месяцам года предлагается идентифицировать 12 регрессионных зависимостей вида

$$P_k = g_k(\bar{P}, P_{k-1}), \quad k = 1, 2, \dots, 12. \quad (2.41)$$

Это выражение показывает, что сток каждого месяца коррелирован со стоком предыдущего и среднегодовым. Неизвестные функции  $g_k$  ищутся в классе полиномов метолами обычного регрессионного анализа, т. е. неизвестные коэффициенты полиномов определяются по методу наименьших квадратов.

Структура полиномиальных зависимостей (2.38) заранее не задана, поэтому с использованием комбинаторного алгоритма МГУА сначала определяется полный набор полиномов заданной степени для каждого из 12 месяцев по данным наблюдений за стоком. Затем составляются всевозможные сочетания из 12 полиномов так, чтобы в каждом из них присутствовал лишь один полином от каждого набора.

Для всех таких сочетаний вычисляется значение критерия баланса (2.41), для которой значение критерия баланса (2.40) минимум, принимается в качестве искомой прогнозирующей модели среднемесячного стока реки.

В полном объеме изложенный метод трудно реализуем на ЭВМ, поскольку слишком велико время счета при полном переборе вариантов. Для сокращения перебора вводится промежуточный критерий, по которому в наборе для каждого месяца составляетя  $F$  (число «свободы выбора» частных полиномов). Например, при второй степени полное число вариантов модели стока равно  $31^{12}$ , а задача

синтеза модели стока становится реальной лишь при  $F \ll 31$ . В качестве промежуточного может быть взят критерий регулярности или минимума смешения. В программе реализован второй из них.

**Описание параметров.** *Баланс:*

*СТ* — максимальная степень частного полинома;

*М* — длина первой обучающей последовательности;

*М1* — длина второй обучающей последовательности;

*М2* — длина экзаменационной последовательности;

*Т* — глубина прогноза (при заданном среднегодовом прогнозе);

*N1* — число тактов дискретизации цикла процесса, полну критерия частных моделей;

*G* — число отбираемых (лучших) по вспомогательному критерию частных моделей;

*RF[1:N1, 1:M+M1+M2+T]* — данные об изменении среднегодового стока предыдущего и

*СГ[1:M+M1+M2+T]* — данные об изменениях стока предыдущего и

*СМ, СМ1[1:M+M1+M2]* — данные об изменениях стока предыдущего и

*N* — число членов частного описания;

*Р1[0:N, 1:N1, 1:G]* — массив коэффициентов для записи промежуточных (*P2*) и лучших по критерию баланса (*P3*) моделей стока (т. е. наборов из *N1* частных моделей).

**Последовательность работы программы.** После ввода значений управляющих параметров и описания массивов считаются данные об изменении среднегодового стока.

Определение *G* лучших частных моделей для каждого из *N1* месяцев (или каждого цикла) выполняется в цикле. Здесь последовательночитываются данные каждого месяца, формируются массивы для синтеза модели вида (2.41) заданной степени от двух переменных и оптимизируется структура этих моделей с помощью комбинаторного алгоритма.

Оформление по критерию минимума сменения модели пересчитываются по последовательности данных длины *M+M1*. Затем выполняется перебор «моделей стока», т. е. различных сочетаний по *N1* моделей и отбор лучшей по критерию баланса, после чего печатается оптимальная модель для всех тактов.

По полученной модели среднемесечного стока осуществляется контрольное прогнозирование и сравнение результатов на данных экзаменационной последовательности, а также — при заданном среднегодовом прогнозе — прогнозирования среднемесечных значений на *T* лет вперед.

Все вводимые данные и основные результаты снабжаются пояснительным текстом.

**Примечания:** 1. В соответствии с видом модели (2.41) исходные данные вводятся в следующей последовательности: за январь, за февраль, ..., за декабрь.

2. Степень полинома не следует задавать больше 3.

3. При *СТ*  $> 2$  все числа делятся на 1000; вычисленные коэффициенты относятся к переменным в новом масштабе.

ПРОГРАММА 16

```

1. 'ALGOL':DES,MAX,NOS,PAR.,
2. 'BEGIN' INTEGER,ST,T,M,M1,M2,N1,G,,READCT,T,M,M1,M2,N1,G,,'
3. 'PRINT',(12*N1,G,ST,,,'IF'G=1,'THEN','PRINT',','
4. 'X,25DG=1-' KОМТЕРИИ СМЕЩЕНИЯ )''ELSE','PRINT',',/X,2DG=12,
5. 'END' - КРИТЕРИИ БАЛАНСА)'',G,,'PRINT',(18*РАЗДЕЛ.ИХ.ДАННЫХ):'',
```

```

6. 'PRINT'',/ЭН0Б.,15,5Х,ЗНПР.,15,5Х,4Н03.,15,5Х,7НПР03.15)'',
7. М,М1,М2,Т.,'PRINT'',/(13Н01Е3ИРУФЧ,15,5Х,9Н0ДИ0М0В)'',N1.,
8. 'БСКИ'',ARRAY CT/1:M+M1+M2*/I),CM,CM1/1:M+M1+M2/),V,W/1:(
9. M2+T)*N1/),RF/1:N1:1:M+M1+M2/),T/1:G/),'INTEGER'ARRAY'
10. B/1:N1/),READ(RF),,'PRINT'',/(12E10.2)'RF.,'BEGIN' INTEGER'
11. C,F,I,J,K,L,N,0,01,S.,'REAL'0,0W,SG,R,Z,Z1,Ф,N:=(CT+1)*
12. ((CT+2)/2-1.,L:=2**N-1.,'PRINT'',/(26НЛДИА ПОДОГ ПОДИОА N-
13. ,L2)'',N.,'PRINT'',/(23НЧИСЛО ЧАСТИХ ПОДНОМОВ,113)'',L.,
14. 'BEGIN' INTEGER'ARRAY D/1:N+1/),P1/1:N+1,1:G/),,'ARRAY'
15. MD/1:M+M1,1:N+1/),FS,FS1/1:N+1,1:N+2/),P,P0/1:N+1/),P1/0:N,
16. 1:G/),P2,P3/0:N,1:N1/),,'PRINT'',/(12НЧИСЛО СИТЕА:'',,
17. C:='IF'M1=0'THEN'1'ELSE'(M+1)','FOR'I:=1'STEP'1'UNTIL'M+M1+M2
18. 'DO' BEGIN'R:=0.,'IF'I>CT','THEN'CM1/I/):=RF/N1,I-1/),
19. 'FOR'J:=1'STEP'1'UNTIL'N1'DO,'R:=RF/J,I/),,'CT/I/):=R/N1
20. 'EQ.,Z1:=CM1/I/):='IF'M1=0'THEN'CM1/I/2/'ELSE'CM1/I/M+2/),
21. 'FOR'Q1:=1'STEP'1'UNTIL'N1'DO,'BEGIN' FOR'I:=1'STEP'1'UNTIL'
22. M+M1+M2'DO'CMK/I/):=RF/01,I/),,'FOR'I:=1'STEP'1'UNTIL'M+M1
23. 'DO' BEGIN'F:=0.,'FOR'J:=0'STEP'1'UNTIL'CT'DO,'FOR'K:=0'STEP'1
24. 'UNTIL'J'DO' BEGIN'F:=F+1.'MD/1,I,F/):=CT/I/**(J-K)*CM1/I/1)***
25. K'END' END'', 'FOR'I:=1'STEP'1'UNTIL'M2+T'DO'V/Н1*(I-1)+0/),
26. :='IF'I LE M2'THEN'CMK/M+M1+T/)'ELSE'0.,NOS(N+1,1,M,MD,CM,FS),
27. NOS(N+1,M+1,M1,MD,CM,FS1),,'FOR'Q:=1'STEP'1'UNTIL'1'DO'
28. 'BEGIN'DEС(O,1,N+1,2),D/I/):=1..PAK(D,N+1,FS,P),PAK(D,N+1,
29. FS1,FS),,'QW:=0.,'FOR'J:=1'STEP'1'UNTIL'M+M1'DO,'BEGIN'Z:=Q:=0.,
30. 'FOR'K:=1'STEP'1'UNTIL'N+1'DO' BEGIN'Z:=Z+P/J/K)*MD/J,K/),
31. Q:='+P0/V/K)*MD/J,K/)'END'', 'QW:=QW+(Z-Q)**2'END'', 'IF'Q'LE'G
32. 'THEN'F:=Q'ELSE' BEGIN'MAXT,G,R,F),,'IF'QW'GE'Р'THEN'GOTO'
33. ME1'END'', 'Г/Ф/):=QW.,'FOR'K:=1'STEP'1'UNTIL'N+1'DO'D/J/K,F/),
34. :=X/K).ME1:=END'',NOS(N+1,1,M+M1,MD,CM,FS),,'Q:=0.,'FOR'
35. J:=1'STEP'1'UNTIL'M+M1'DO'Q:=Q+CM(J/)**2.,'FOR'J:=1'STEP'1
36. 'UNTIL'G/DO' BEGIN'FOR'K:=1'STEP'1'UNTIL'N+1'DO'D/J/K/):=
37. DI/J,K/),,РАР(D,N+1,FS,P),,'FOR'K:=1'STEP'1'UNTIL'N+1'DO'
38. P1/I-1,O1,J/):=P/(K/),,QW:=QW+(Z-Q)**2'END'', 'IF'Q'LE'G
39. 'DO' BEGIN'Z:=0.,'FOR'K:=1'STEP'1'UNTIL'N+1'DO'Z:=Z+P/V/K)*
40. MD/V,1,W/),,QW:=QW+(CM/V/I-Z)**2'END'', 'Г/V/):=SORT(QW/Ф)

```

### Глава 3

#### СТРУКТУРНАЯ ИДЕНТИФИКАЦИЯ АЛГЕБРАИЧЕСКИХ И КОНЕЧНО-РАЗНОСТНЫХ МОДЕЛЕЙ

##### 1. ЗАДАЧА СТРУКТУРНОЙ ИДЕНТИФИКАЦИИ

Пусть для идентификации объекта или процесса задан ряд точек (результаты наблюдений за изменением входных и выходных параметров). Требуется определить модель (математическое описание) объекта (процесса), в случае,

когда структура объекта и соответственно модели неизвестна или известна неполностью. Для решения этой задачи, т. е. одновременного определения оптимальной структуры и оценки параметров модели, «генерируются» определенные наборы моделей различной сложности и отбираются (секретируются) лучшие из них по целесообразно заданному критерию регуляризации.

Все алгоритмы МГУА различаются на две большие группы: многорядные и комбинаторные. В данной главе рассматриваются алгоритмы идентификации, относящиеся как к однотипной, так и к другой группе, а также смешанные многорядно-комбинаторные алгоритмы. Все они предназначены для идентификации объектов в классе моделей с сопредоточенными параметрами и имеют ряд общих признаков.

**Структура алгоритмов идентификации.** Процесс синтеза модели состоит из трех последовательных этапов (или фаз): определения структуры, оценки параметров и адаптации коэффициентов.

При определении структуры выбирается модель оптимальной сложности по зашумленным данным наблюдений за входами и выходами объекта, так и при оптимальности понимается минимум некоторого целесообразно называемого критерия, выбранного автором модели и учитывающего ее назначение (прогноз, идентификация, управление и т. п.). Рекомендуется применять критерии минимума смешения или комбинированные критерии, которые позволяют найти модель оптимальной сложности как в случае постоянного закона объекта, так и при медленном изменении его структуры и параметров.

При оценке параметров или коэффициентов модели обычно применяют метод наименьших квадратов. Такие оценки соответствуют минимуму среднеквадратической ошибки, посчитанной на всех точках таблицы исходных данных. Оценки могут быть улучшены, если более точно учесть распределение помех.

На этапе адаптации коэффициентов структура модели не изменяется. Оценки коэффициентов, полученные по методу наименьших квадратов, применяются за исходное (первое) приближение и уточняются (повышается точность экстраполяции или прогноза). На этапе адаптации коэффициентов применяется случайный поиск в пространстве коэффициентов модели, градиентные методы или методы стохастической аппроксимации. Для однозначности выбора оценок коэффициентов поиск ведется по комбинированым критериям (см. гл. 2).

После окончания идентификации, оценки параметров и этапа адаптации коэффициентов модель готова к применению для решения задач прогнозирования, распознавания образов или оптимального управления. В последнем случае решается еще и задача *синтеза оптимального управления* (которую можно рассматривать как некоторый четвертый этап алгоритма синтеза). При этом к модели предъявляются некоторые специфические требования, например, наличие управляющих воздействий среди аргументов модели.

**Алгебраические и конечно-разностные модели.** В данной главе собраны программы, которые пригодны для синтеза как конечно-разностных, так и алгебраических моделей (при соответствующем представлении исходных данных).

**Конечно-разностные модели являются дискретными аналогами дифференциальных моделей, которые обеспечивают более разнообразный набор решений, чем алгебраические модели и, следовательно, более высокую точность прогнозов при большем времени упреждения. Преимущество конечно-разностных моделей состоит также в том, что их решение получается при помощи пошагового интегрирования, если невозможно аналитическое решение.**

При идентификации алгебраических моделей, т. е. моделей статистических объектов или устанавлившихся процессов, в таблице исходных данных содержатся выходная величина и входные аргументы, измеренные в один и тот же момент времени. При этом исходную модель представляют в следующем виде:

$$q = f(X_1, X_2, \dots, X_M), \quad (3.1)$$

где часть аргументов может содержать управляющие воздействия и время (изменяющееся дискретно). Модель вида (3.1) применима для интер- и экстраполяции, а также управления.

Для идентификации динамических объектов или процессов «с памятью» модели синтезируются в классе конечно-разностных уравнений. При этом таблица

данных дополняется «запаздывающими» значениями всех аргументов (в том числе выходных переменных) на 1, 2 и т. д. тактах шага. В простейшем случае при одном управляемом и одном возмущающем (замеряющем)  $\lambda$  воздействии конечноразностная модель имеет вид

$$\dots, \lambda_{+1}, \lambda_0, \lambda_{-1}, \lambda_{-2}, \dots, t). \quad (3.2)$$

Такая модель может служить для синтеза оптимального управления, а также для прогнозирования.

Число запаздывающих аргументов модели вида (3.2), а также структура оптимальной сложности моделей вида (3.1) и (3.2) находятся по алгоритмам саморегуляции. Если в модели (3.2) все переменные представить в виде обобщенного вектора аргументов  $X'_1, X'_2, \dots, X'_M$ , то задачи синтеза моделей вида (3.1) и (3.2) окажутся идентичными, т. е. их можно решать с применением одинаковых же алгоритмов МГУА.

**Краткая характеристика программ структурной идентификации моделей.** В гл. 3 представлены пять программ, отличающихся в основном способами «секретирования» частных описаний, а также применяемыми критериями и назначением. Одна из программ (четвертая) является специализированной и позволяет идентифицировать динамические объекты управления. Остальные программы применимы для синтеза как алгебраических, так и конечно-разностных моделей.

Первая программа представляет собой «классический» многорядный алгоритм МГУА с селекцией поларных частных описаний (линейных и нелинейных). В основе второй программы лежат полные селекции на каждом ряду одного человека, который ищется в виде произведения различных аргументов и их обобщенных аргументов (членов), поэтому каждый раз определяется лишь один коэффициент.

Третья программа является комбинаторной, т. е. использует полный перебор частных моделей, содержащих по 1, 2, 3 и т. д. членов модели максимальной сложности, и селекцию лучших из них по заданному критерию.

В четвертой программе применяется быстрая многорядная селекция поларных частных описаний, но для управляющих воздействий селекция осуществляется независимо.

В пятой программе общая многорядная структура дополнена комбинаторной оптимизацией сложности каждого частного описания.

В качестве критерия селекции применяются как описанные в гл. 1 и 2, так и некоторые специфичные критерии, охарактеризованные в отдельных программах.

## 2. ПРОГРАММА ИДЕНТИФИКАЦИИ ХАРАКТЕРИСТИК СЛОЖНЫХ ОБЪЕКТОВ ПО МГУА

Программа (прогр. 17) предназначена для идентификации характеристик статистических и динамических объектов [27]. Полученные модели могут использоваться как для целей управления, так и для краткосрочного прогнозирования.

Язык программирования АЛГОЛ, ЭВМ БЭСМ-6.

**Используемые программы и функции.** FORMXPI, GRAF, NEML, У, УУ, НУ, КОЭФ, SQRТ, ABS.

**Метод.** Пусть имеется  $n$  входных параметров и одна выходная величина.

Необходимо найти связывающее их уравнение полиномального вида.

При идентификации характеристик модель имеет следующий вид:

$$\text{для статических объектов} \quad \hat{\Phi}_t = f((x)_t), \quad (3.3)$$

$$\hat{\Phi}_t = f((x)_t, \{x\}_{t-1}, \dots, \{x\}_{t-n}, \Phi_{t-1}, \dots, \Phi_{t-n}). \quad (3.4)$$



промежуточных переменных предпоследнего ряда в исходном порядке следования появляются для всех моделей (если  $TU > 0$ , первые точки промежуточных переменных соответствуют  $TU + 1$ -й точке выходной величины) (строки 80, 82, 85); значения коэффициентов моделей и критериев отбора при  $L = 3$  выводятся; значение критерия отбора и СКО для всех моделей и рядов (строки 88—93), а также значения коэффициентов развернутых уравнений (исключены все промежуточные переменные) (строка 93); при  $L \geq 3$  выводится: номер ряда, индексы пар, значения коэффициентов промежуточных переменных, СКО и значение критерия (строки 94—97).

Таблица 3

**Индексы и коэффициенты промежуточных переменных**

Ряд	Пара		Коэффициенты промежуточных переменных						SIGMA	Коэффициент корреляции
	$i$	$j$	$a_{ij}^1$	$a_{ij}^2$	$a_{ij}^3$	$a_{ij}^4$	$a_{ij}^5$	$a_{ij}^6$		
1	1	2	$a_{01}^1$	$a_{11}^1$	$a_{21}^1$	$a_{31}^1$	$a_{41}^1$	$a_{51}^1$	$\sigma_1^1$	$K_1^1$
	2	3	$a_{02}^1$	$a_{12}^1$	$a_{22}^1$	$a_{32}^1$	$a_{42}^1$	$a_{52}^1$	$\sigma_2^1$	$K_2^1$
	3	1	$a_{03}^1$	$a_{13}^1$	$a_{23}^1$	$a_{33}^1$	$a_{43}^1$	$a_{53}^1$	$\sigma_3^1$	$K_3^1$
	1	3	$a_{01}^2$	$a_{11}^2$	$a_{21}^2$	$a_{31}^2$	$a_{41}^2$	$a_{51}^2$	$\sigma_1^2$	$K_1^2$
	2	1	$a_{02}^2$	$a_{12}^2$	$a_{22}^2$	$a_{32}^2$	$a_{42}^2$	$a_{52}^2$	$\sigma_2^2$	$K_2^2$
	3	2	$a_{03}^2$	$a_{13}^2$	$a_{23}^2$	$a_{33}^2$	$a_{43}^2$	$a_{53}^2$	$\sigma_3^2$	$K_3^2$
2	1	3	$a_{01}^3$	$a_{11}^3$	$a_{21}^3$	$a_{31}^3$	$a_{41}^3$	$a_{51}^3$	$\sigma_1^3$	$K_1^3$
	2	1	$a_{02}^3$	$a_{12}^3$	$a_{22}^3$	$a_{32}^3$	$a_{42}^3$	$a_{52}^3$	$\sigma_2^3$	$K_2^3$
	3	2	$a_{03}^3$	$a_{13}^3$	$a_{23}^3$	$a_{33}^3$	$a_{43}^3$	$a_{53}^3$	$\sigma_3^3$	$K_3^3$
	1	2	$a_{01}^4$	$a_{11}^4$	0	$a_{31}^4$	0	0	$\sigma_1^4$	$K_1^4$
	2	3	$a_{02}^4$	$a_{12}^4$	0	$a_{32}^4$	0	0	$\sigma_2^4$	$K_2^4$
	3	1	$a_{03}^4$	$a_{13}^4$	0	$a_{33}^4$	0	0	$\sigma_3^4$	$K_3^4$
3	1	2	$a_{01}^5$	$a_{11}^5$	0	$a_{31}^5$	0	0	$\sigma_1^5$	$K_1^5$
	2	3	$a_{02}^5$	$a_{12}^5$	0	$a_{32}^5$	0	0	$\sigma_2^5$	$K_2^5$
	3	1	$a_{03}^5$	$a_{13}^5$	0	$a_{33}^5$	0	0	$\sigma_3^5$	$K_3^5$

ки 95—100). Ниже иллюстрируется расшифровка полученных результатов при  $L \geq 4$ .

Пусть на печать выведена информация, представленная в табл. 3 ( $L = 4$ ). Причем  $K_{\max}^2 > K_{\max}^4$ ,  $K_{\max}^3 < K_{\max}^2$  и  $K_{\max}^2 = K_1^2$ , тогда окончательное решение имеет вид

$$z_{13} = a_{01}^2 + a_{11}^2 y_{12} + a_{21}^2 y_{31} + a_{31}^2 y_{42},$$

где  $y_{12} = a_{01}^1 + a_{11}^1 x_1 + a_{21}^1 x_2 + a_{31}^1 x_3 + a_{41}^1 x_4$ .

**Пример 3.** Решается задача идентификации характеристик объекта, исходные данные (входные аргументы) которого приведены в табл. 4 (в последней шестнадцатой строке занесены значения выходной величины);  $N = 5$ ,  $N_1 = 16$ ,  $K = 10$ ,  $K_1 = 6$ ,  $L = 6$ . Наилучшая модель (по максимуму коэффициента корреляции) получена во втором ряду:

$$\hat{\phi} = z_{14} = 67.99 + 19.67Y_{12} + 1.349Y_{41},$$

где  $Y_{12} = -1.289 + 0.1707X_1 + 0.0253X_2 + 0.00317X_1X_2 - 0.0074X_1^2 -$

$$-0.000444X_2^2$$

$$Y_{41} = -0.714 + 0.00011X_4 + 0.131X_1 + 0.00004X_1X_4 -$$

$$-0.0000006X_4^2$$

**Примечания:** 1. Структурная таблица (см. с. 59) позволяет выбрать из всех рабочих аргументов наиболее существенные, т. е. те, которые наиболее часто участвуют в образовании моделей.

Таблица 4

**К расчету примера 3**

No	$X_1$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\varphi$	No	$X_1$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\varphi$
1	6	20	36	400	72	9	8	70	64	4900	88	
2	6	30	36	900	76	10	8	40	64	1600	87	
3	5	45	25	2025	75	11	7	20	49	400	74	
4	4	50	16	2500	59	12	4	30	16	900	68	
5	4	40	16	1600	70	13	5	30	25	1600	72	
6	5	60	25	3600	73	14	6	40	36	79	78,5	
7	5	50	25	2500	75	15	8	40	36	1600	79	
8	6	70	36	4900	76	16	8	50	64	2500	89	

2. Без расширения памяти (использования барабанов или ленты для записи промежуточных переменных) можно решать задачи с  $N^* < 50$  и  $N_1^* < 20$  (при уменьшении числа аргументов число точек может быть увеличено).

3. При задании длины обучающейся и проверочной последовательностей можно пользоваться следующими соотношениями:  $K \leq 0.6 (N_1^* - TU)$ ,  $K_1 \leq 0.4 (N_1^* - TU)$ ,  $K + K_1 \leq N_1^* - TU$ .

Точки от  $N_1^* - TU + 1$  до  $N_1^* - TU$  составляют экзаменационную последовательность. В синтезе модели они не участвуют, а служат для проверки адекватности модели объекту. Точки от  $N_1^* - TU + 1$  до  $N_1^* - TU$  — прогнозные значения выходной величины.

ПРОГРАММА 17

```

1. 'ALGO1':NAME,GRAF,Y,IW,UW,KOEF,FORMXII,
2. 'BYLIPR',INTEGER,N,N1,K,K1,L,T3,TY,TC,,READIN,N1,K,K1,L,
3. T3,IW,TC),N:=I*(T3+1)*TC+(IF TU,EQ'0'THEN'0'ELSE'1'),
4. N1:=N1-T3,,,'PRINT',,(3N=N,,13,10X,3N1=,13,10X,2Nk=,
5. T3,10X,3N1=,13,10X,2Nk=,72,10X,3N1=,13,10X,3N1=,
6. 13,10X,3N1=,13,10X,2Nk=,,N,N1,K,K1,L,T3,TY,TC,
7. 'BEGIN'ARRAY'X(1:N,1:IW'),XS(1:2,1:W'),XL,XP(1:N,1:N1),
8. D2(I:N-1,1:N-1),'INTEGER,J,I,H,C,F,R,T,,REAL,DK,D2M,
9. 'INTEGER','ARRAY'IK/1:N-1,1:N-1/,W/1:N1/,
10. FORMXII(X,N,M,T3,TY,TC),,'FOR',I:=1'STEP'1'UNTIL'N'DO',
11. ('BEGIN'X$/(1,L):=XS(2,1):=X((1,1),,'FOR',J:=2'STEP'1
12. 'UNTIL'N'DO,'IF'XS((1,1):=LT'X(J,1)'THEN'XS((1,1):=
13. X(J,1),'ELSE','IF'XS((2,1):=GT'X(J,1)'THEN'XS((2,1):=

```

```

14. X(J,I)/*END*/. FOR I=1 STEP 1 UNTIL N DO /* BEGIN
15. DK:=XS(J,1,L)-XS(J,2,I)/., 'FOR' J:=1 STEP 1 UNTIL N1 DO'
16. X(J,I):=(X(J,I)-XS(J,2,I))/DK /* END */,
17. HL/* BEGIN */ ARRAY S/(1:I+1,1:L+1/), X1(/1:L+1,1:N1/),
18. A2(/1:L+1/), HU(/1:N-1,1:N-1/). /* BEGIN */ REAL W., 'FOR' I:=1 STEP 1.
19. 'ARRAY' L0(/1:N-1,1:N-1/). /* BEGIN */ REAL W., 'FOR' I:=1 STEP 1.
20. 'UNTIL' N DO /* BEGIN */ XH(/1,1/):=0., 'FOR' J:=1 STEP 1 UNTIL N1
21. 'DO' XK(/1,1/):=XH(/1,1/)*X(/J,1/). XK(/1,1/):=XH(/1,1/)*N1
22. 'END' ., 'FOR' I:=1 STEP 1 UNTIL K+K1 DO X1(/1,I/):=0., 'FOR' I:=1
23. 'STEP' 1 UNTIL K+K1 DO /* BEGIN */ W:=0., 'FOR' J:=1 STEP 1 UNTIL
24. N-1 DO W:=W+(X(/1,J/)-X(/J,1/))*2., 'FOR' J:=1 STEP 1 UNTIL
25. I DO /* IF W > X1(/1,J/) THEN /* BEGIN */ FOR H:=I-1 STEP -1
26. 'UNTIL' J DO /* BEGIN */ X1(/1,H/):=X1(/1,H/)+W/(H+1), 'WHICH/H'
27. 'END' ., GOTO B12 /* END */. B12:X1(/1,J/):=W. H(/J/):=I /* END */
28. H:=(K+K1)/2, R:=H-1., 'FOR' I:=2 STEP 2 UNTIL H DO /* BEGIN */
29. J:=W(/1/), G:=R+I, W(/1/):=W(G/), H(G/):=J /* END */,
30. 'FOR' I:=1 STEP 1 UNTIL K+K1 DO /* BEGIN */ FOR J:=1 STEP 1 UNTIL N DO
31. XH(/J,1/):=X(W(/1/),J) /* END */. IF K+K1 LT N1 THEN
32. 'FOR' I:=K+K1+1 STEP 1 UNTIL N1 DO /* BEGIN */ W(/1/):=I,
33. 'FOR' J:=1 STEP 1 UNTIL N DO XH(/J,1/):=X(I,J) /* END */,
34. W1:=0., 'FOR' I:=K+1 STEP 1 UNTIL K+K1 DO W1:=W1+XP(N,I)**2.,
35. AT:='FOR' T:=1 STEP 1 UNTIL N-1 DO /* BEGIN */ FOR H:=1 STEP
36. 1 UNTIL N-1 DO IH(/T,H/):=0., 'FOR' I:=1 STEP 1 UNTIL N-1 DO
37. / BEGIN D2(/T,I/):=-1., 'FOR' J:=1 STEP 1 UNTIL N-1 DO
38. 'BEGIN' IF I*T J ED J*IH(T,J) AND J LT I THEN GOTO H.,
39. YY(X1,XP,I,J,L,N1),, H(S,X1,1,K,L),, /* BEGIN */ REAL D1.,
40. H:=0, NEMLL,S,A2,X1,K,K+K1,DI,W1,H,, 'IF' D2(T,I) LT D1
41. 'THEN' /* BEGIN */ D2(/T,I/):=D1.. IH(V,T,L):=J, LO(V,T,I):=H /* END */
42. 'END' ., J:/* END */ END ., 'FOR' J:=1 STEP 1 UNTIL N-1 DO /* BEGIN */
43. 'REAL' D1, YY(X1,XP,J,IH(V,T,J)), L,N1),, H(S,X1,1,K+K1,L),
44. NEML1,S,A2,X1,1,K+K1,DI,W1,LO(V,T,J)), Y(L,J,N1,X1,A2,X1),
45. 'END' ., J:/* END */ END ., 'FOR' J:=1 STEP 1 UNTIL N-1 DO /* BEGIN */
46. D2M:=-1., 'FOR' H:=1 STEP 1 UNTIL N-1 DO /* IF D2M LT D2(V,T,H) */
47. 'THEN' D2M:=D2(/T,H/),, 'IF' T=1 THEN /* BEGIN */ DK:=D2M,
48. 'GOTO' M1 /* END */. 'IF' D2M=DK LE .0001 THEN GOTO H, DK:=D2M,
49. 'IF' T=N-1 THEN GOTO M.. M1:='FOR' H:=1 STEP 1 UNTIL N-1 DO
50. 'FOR' F:=1 STEP 1 UNTIL N1 DO /* BEGIN */
51. E:/* BEGIN */ INTEGER, ARRAY IN(1:2,1:N/), 'STRING', ARRAY
52. II(/1:N/),, 'FOR' I:=1 STEP 1 UNTIL N-1 DO /* BEGIN */ 'FOR' H:=1
53. 'STEP' 1 UNTIL N DO IH(/2,H/):=0., IN(2,I):=I.. G:=2.,
54. 'FOR' H:=2 STEP 1 UNTIL L DO /* IF IH(/T-1,IH/)>0 THEN */
55. G:=G+1., 'IF' G NE L THEN IH(/2,H/):=IH(/T-1,IH/),
56. 'FOR' J:=T-2 STEP -1 UNTIL 1 DO /* BEGIN */ 'FOR' H:=1 STEP 1 UNTIL
57. N DO /* IF IH(/J,H,F/)>0 THEN G:=G+1., 'IF' G NE L THEN */
58. IF IH(/J,H/):=N THEN /* BEGIN */ G:=2., 'FOR' F:=3 STEP 1 UNTIL
59. L DO /* IF IH(/J,H,F/)>0 THEN G:=G+1., 'IF' G NE L THEN */
60. IH(/2,IH(J,H/)):="N", IH(/J,H/):=END ., 'FOR' H:=1 STEP 1
61. 'UNTIL' N DO /* IF IH(/H/):=( 'IF' IH(/2,H/)>0 THEN */
62. 'ELSE' /* I */ ., PRINT(NEWLINE, H) /* END */. PRINT(NEWLINE)
63. 'FOR' I:=1 STEP 1 UNTIL N DO /* IF IH(/1,I/):=XS(1,I) THEN */
64. 'FOR' I:=1 STEP 1 UNTIL N-1 DO /* BEGIN */ SWITCH P:=I3,I4,I5,
65. IH(/1,I,2/):=DK:=IH(/1,1,2/)/XS(1,I), IH(/1,1,3/):=
66. IH(/1,1,3/)/XS(1,IH(1,1,1/)), IX1:=IH(/1,1,3/), IH(/1,1,1/),
67. :=IH(/1,1,1/)-DK*XS(2,I)-DK*XS(2,IH(/1,1,1/)), 'GOTO'
68. P(/I-2/), IH(/1,1,6/):=DK:=IH(/1,1,6/)/XS(1,IH(1,1,1/))
69. /*2., IH(/1,1,1/):=IH(/1,1,1/)+IX1/XS(2,IH(1,1,1/))**2.,
70. IH(/1,1,3/):=IH(/1,1,3/)-2*IX1/XS(2,IH(1,1,1/)),
71. IX1:=IH(/1,1,5/):=DK:=IH(/1,1,5/)/XS(1,IH(1,1,1/))**2., IH(/1,1,1/):=
72. IH(/1,1,1/)+DK*XS(2,IH(1,1,1/))**2., IH(/1,1,2/):=IH(/1,1,2/)-2*DK
73. XS(2,I), I4:=IH(/1,1,4/):=DK:=IH(/1,1,4/)/(XS(1,IH(1,1,1/))
74. XS( V,IH(1,1,1/)).. IH(/V,1,1,1/):=IH(/V,1,1,1/)+DK*XS(2,IH(1,1,1/))
75. XS(2,IH(1,1,1/)), IH(/V,1,1,2/):=IH(/V,1,1,2/)-DK*
76. XS(2,I), IH(1,V), .., IH(V,1,1,3/):=IH(V,1,1,3/)-DK*XS(2,IH(1,1,1/))
77. I3:/* OR */ /*1 STEP 1 UNTIL L DO /* IF IH(/T-1,IH/):=IH(V,T-1,IH/)*
78. XS(1,N/), IH(V,T-1,IH/):=IH(V,T-1,IH/)+XS(2,N/)/END .,
79. 'PRINT', /* ЭВОЛЮЦИЯ ВЫХОДНОЙ ВЕЛИЧИНЫ ( Н МОДЕЛЬ ) */,
80. 'FOR' I:=1 STEP 1 UNTIL T DO /* FOR J:=1 STEP 1 UNTIL N-1 DO */
81. 'BEGIN' D2M:=IH(V,T-1,IH/)(K+K1),, IH(V,T-1,IH/):=( 'IF' D2M LT
82. 0 THEN SORT(D2M), ELSE -SORT(-D2M)*XS(1,N/),, 'FOR' H:=1
83. 'STEP' 1 UNTIL N-1 DO /* BEGIN */ FOR G:=1 STEP 1 UNTIL N1 DO
84. X1(/1,IH(G/)):="XP(V,H,G/),, PRINT((8(E14,8,2X))),, 'FOR' G:=1
85. 'STEP' 1 UNTIL N1 DO /* GOTO(X1(/1,G/)*XS(1,N/)+XS(2,N/)) */ END .,

```

```

86. G:=0, 'FOR' I:=1'STEP'1'UNTIL'T'DO' 'FOR' J:=1'STEP'1'UNTIL'N-1
    'DO' 'IF' I<J 'THEN' G:=G+1, 'IF' G>0' THEN' 'BEGIN'
88. 'PRINT', '(45H ВВОД ЗНАЧЕНИЙ К.КОР./СКО ПО РЯДАМ ДЛЯ ВСЕХ',
89. 'ЧИСЕЛЕЙ)', ', 'PRINT', '(25X,ЗНЧД,(5X,10(X,I2,2H-И,4X)))',
90. 'FOR' I:=1'STEP'1'UNTIL'T'DO' I, 'FOR' I:=1'STEP'1'UNTIL'N-1'DO'
    'BEGIN' 'PRINT', '(15,10E12.5)', I, 'FOR' J:=1'STEP'1'UNTIL'T'DO'
91. 'D2(J,I)', 'PRINT', '(7X,10E12.5)', ', 'FOR' J:=1'STEP'1'UNTIL'T
92. 'D2(J,I)', 'PRINT', '(7X,10E12.5)', ', 'FOR' J:=1'STEP'1'UNTIL'T
93. 'DO' 'KIN(J,I,L+I)' END', 'КОФФИК.Н.Т,TH,XS)' END' ELSE'
94. ('BEGIN' 'INTEGER' I,J,H,, 'ARRAY' A(/1:6/), 'FOR' I:=1'STEP'1,
    'UNTIL'6'DO'A(/I/):=0, 'PRINT', '(4X,4H)A,12X,
95. 12НКОЭФФИЦИЕНТЫ,75X,21HSIGMA   К.КОР.ПР.)', ,
96. 'FOR' I:=1'STEP'1'UNTIL'T'DO' ('BEGIN' 'PRINT', '(ЭНРНД,15)', I,
97. 'FOR' J:=I'STEP'1'UNTIL'N-1'DO' ('BEGIN' 'FOR' H:=1'STEP'1'UNTIL'
98. 'FOR' J:=I'STEP'1'UNTIL'N-1'DO' ('BEGIN' 'FOR' H:=1'STEP'1'UNTIL'
99. 1'DO'A(/H/):=KIN(I,J,H), 'PRINT', '(214,6(E14.8,2X),E12.6,2X,
100. F8.5)', J, IN/(I,J/), A, KIN(I,J,L+I/), D2(I,J/)'END', END', END'.
101. GRAFXP,N,N1,I,K+1,XS(/2,N/),XS(/1,N/),TY)
102. ,END', END', END', EOP'

```

В программе используются следующие подпрограммы:

#### ПОДПРОГРАММА FORMXH1

Формирование рабочего массива данных из исходного вводимого массива (прогр. 18)

*Обращение.* FORMXH1 (X, N, N1, T3, TY, TC)

*Описание параметров.*

X — выходной массив данных (рабочий массив исходных данных) размерности N на N;

N — число точек наблюдения;

T3 — учитываемое число интервалов западывания;

TY — количество интервалов упреждения;

TC — ключ.

*Преуемые функции и подпрограммы.* Нет

*Метод.* Массив данных представляет собой  $n + 1$  подмассивов ( $n$  — число аргументов исходного массива). Каждый  $i$ -й ( $i = \overline{1, n}$ ) подмассив — значения аргументов;  $n + 1$ -й подмассив — значения выходной величины, записанной в порядке ее естественного следования (выходной величине соответствует последний подмассив). Ввод данных производится последовательно, подмассивы, одновременно с этим производятся распечатка подмассивов, одновременно с этим производится распечатка подмассивов подмассивов. Соответствие между аргументами рабочего массива  $X^*$  и массива исходных данных X устанавливается выражениями:

$S(X_{\Lambda,h+B})$  при  $\iota = TC \cdot \gamma + 1$ ,  $TC = 2$ ,  $\iota \leq n \cdot (T3 + 1)$ ;  
 $X_{\Lambda,h+B}$  при  $(\iota = TC \cdot \gamma \text{ и } TC = 2)$  или  $TC = 1$ ,  $\iota \leq n \cdot (T3 + 1)$ .

$S(\Phi_{n+\beta})$  при  $i = TC \cdot \gamma + 1$ ,  $TC = 2$ ,  $n \cdot (T3 + 1) < \iota \leq$   
 $\leq (n + 1) \cdot (T3 + 1)$ ;

$$\begin{cases} \Phi_{n+\beta} & \text{при } (i = TC \cdot \gamma \text{ и } TC = 2) \text{ или } TC = 1, \quad n \cdot (T3 + 1) < \\ & < i \leq (n + 1) \cdot (T3 + 1); \\ \Phi_{h+Ty} & \text{при } i = (n + 1) \cdot (T3 + 1) + 1, \end{cases}$$

где  $X, X^*$  — аргументы исходного и рабочего массивов соответственно;  $\Phi$  — выходная величина исходного массива;  $\Lambda = [i + T3]/(T3 + 1)$ ;  $B = \frac{i - (A - 1)}{1, (n + 1) \cdot (T3 + 1)}$ ;  $\Phi = \frac{\text{SIGN}(Ty)}{1 + \text{SIGN}(Ty)}$ ,

$$\text{SIGN}(Ty) = \begin{cases} 1 & \text{при } Ty > 0; \\ 0 & \text{при } Ty = 0; \\ -1 & \text{при } Ty < 0, \end{cases}$$

$$\Sigma(X_{\Lambda,h+B}) = \begin{cases} 0 & \text{при } h = 1; \\ \sum_{k=1}^{h-1} X_{\Lambda,k+B} & \text{при } h > 1; \end{cases}$$

$$h = [1, n_1 \rightarrow T3]; n_1 \text{ — число точек наблюдений в исходном массиве};$$

$$\Phi_{h+Ty} = \begin{cases} \Phi_{h+Ty} & \text{при } h + Ty \leq N; \\ \Phi_{N1} & \text{при } h + Ty > N. \end{cases}$$

#### ПРОГРАММА 18

```

1. 'PROCEDURE FORMH1(X,N,N1,T3,TY,TC),
2.   'ARRAY X,,,' INTEGER'N,M1,T3,TY,TC.,
3.   'BEGIN' 'INTEGER' I,J,H,G,P,P1..P1:=N1+T3.,
4.   P:='IF' TY>0' THEN' N-1'ELSE' N-1)/(TC*(T3+1)),,
5.   'BEGIN' 'ARRAY' BX(/1:P1/),G:=0,, 'FOR' I:=1'STEP'1'UNTIL'P'DO'
6.   'BEGIN' REALX BX,,,'PRINT', '(16F8.2)', BX, 'FOR' J:=0'STEP'1
7.   'UNTIL' T3'DO' 'BEGIN' G:=G+IC,, 'IF' TC>0' THEN' 'BEGIN'
8.   X(/1,G-1/):=0., 'FOR' H:=2'STEP'1'UNTIL'N1'DO' X(/H,G-1/):=
9.   X(/H-1,G-1/)+BX(/H+J-1/)'END', 'FOR' H:=1'STEP'1'UNTIL'N1
10. 'DO' X(/H,G/):=BX(/H+J/)'END', 'IF' TY>0' THEN'
11. 'GOTO' L1..;:=K+1..;:=TY+T3., 'FOR' I:=1'STEP'1'UNTIL'G'DO'
12. X(/I,N/):=BX(/I+J/), 'FOR' J:=G+1'STEP'1'UNTIL'N1'DO'
13. X(/J,N/):=BX(/N1/),,L1:'END', END', EOP'

```

#### ПОДПРОГРАММА NEML

Решение системы линейных уравнений с выбором оптимальной длины частного описания и вычисление значения критерия отбора (прогр. 19).

*Обращение.* NEML (L; S, A2, X1, K, KK, D1, W1, M).

*Описание параметров:*

S — вводимая матрица нормальных уравнений, размерности  $L+1$  на  $N+1$ ;

X1 — вводимая матрица условных уравнений, размерности  $L+1$  на  $N+1$ ;

A2 — выводимый вектор решений длиной  $L + 1$  ( $L + 1$ -й элемент вектора — сумма квадратов отклонений выходной величины модели и объекта на  $K$  точках);

$L$  — максимальная длина частного описания;

$K$  — длина обучающей последовательности;

$KK$  — общая длина обучающей и проверочной последовательностей;

$D1$  — значение критерия отбора (коэффициента корреляции);

$W1$  — сумма квадратов функций на точках проверочной последовательности;

$M$  — к-тюч. При  $M = 0$  производится оптимизация длины частного описания при  $N > 0$  — оптимизация не производится.

*Требуемые функции и подпрограммы.* SQRT, ABS.

$H$  (новая система полиномов Чебышева) [32] за  $t$  этапов ( $t$  — длина частного описания). В процессе последовательного эллиптизирования вычисляются (уточняются) коэффициенты, вычисляются оценки вновь вводимого  $i$ -го аргумента через все

ранее введенные  $x_i = K_{0i} + \sum_{j=1}^{t-1} K_{jix_j}$ , а также суммы квадратов отклонений значений этой оценки от действительных значений аргументов  $\sum_{i=1}^k x_{i,-1,\dots,0}^2 = \sum_{h=1}^K (x_{i,h} -$

$- \hat{x}_{i,h})^2$  ( $i = \overline{1, t}$ ,  $t$  — длина частного описания). Если  $\sum_{i=1}^k x_{i,-1,\dots,0}^2 / \sum_{h=1}^K x_h^2 < \varepsilon$ ,  $i$ -й аргумент почти линейно выражается через ранее введенные аргументы (в приведенном примере  $\varepsilon = 0,001$ ). Для продолжения счета  $i$ -й аргумент необходимо исключить (произвольится обнуление  $i$ -й строки матрицы  $S$ ). Начиная с третьего этапа эллиптизации оптимизируется длина частного описания — по данным проверочной последовательности вычисляются значения критерия отбора (коэффициента корреляции (2.3)). В частности оптимальный выбирается такая длина частного описания (параметр  $M$ ), для которой значение критерия максимальное.

#### ПРОГРАММА 19

```

1.  'PROCEDURE'NEM(L,S,A2,X1,K,KK,D1,W1,M,,)
2.  'ARRAY'S,A2,X1,,REAL'D1,,W1,,INTEGER'K,KK,M,L,,'
3.  'BEGIN'ARRAY'A(/1:L+1:1:L/)'A1(/1:L+1,1:3/),,,'
4.  ,INTEGER'I,J,H,G,,REAL'S1,F1,W,D,,DL:=-1,,'
5.  ,FOR'I:=1'STEP'1'UNTIL'D1DO'A(/I,1,I/):=A(/I+1,I/):=0,,,
6.  ,FOR'I:=2'STEP'1'UNTIL'L+1'DO'BEGIN'A(/I,1/):=S(/I,1)/'
7.  ,S(/1,1/),,A1(/1,1/):=S(/1,1/)-A(/1,1/)*S(/1,1/)'END'',
8.  ,A(/V,1,1/):=1/S(/V,1,1/),,FOR'J:=2'STEP'1'UNTIL'L+1'DO'ZEGIN'
9.  ,IF'S(/J,J/)'EQ'0'THEN'GOTO'E,,'
10. ,IF'A1(/J,1/)'EQ'0'THEN'GOTO'E,,'
11. ,FOR'I:=J+1'STEP'1'UNTIL'L+1'DO'BEGIN'S1:=0,,'
12. ,FOR'H:=1'STEP'1'UNTIL'J-1'DO'S1:=S1+A(/J,H)*S(/V,H/),,'
13. ,A1(/V,1,2/):=S(/V,1,2/)-S1..A(/V,1,2/):=A(/V,1,2/)/A1(/J,1/),,,'
14. ,FOR'H:=1'STEP'1'UNTIL'J-1'DO'A(/V,H/):=A(/V,H/)-A(/J,H/)*
15. A(/V,1,2/),,A1(/V,1,2/):=A(/V,1,2/)-A(/V,1,2/)*A1(/V,1,2/)'END'',
16. ,FOR'H:=1'STEP'1'UNTIL'J-1'DO'A(/H,H/):=A(/H,H/)-A1(/J,1/),,,'

```

#### ПОДПРОГРАММА КОЭФ

Исключение промежуточных переменных и получение развернутого уравнения для линейных частных описаний (прогр. 20).

*Обращение.* КОЭФ (К1, N, T, IH).

*Описание параметров:*

КП — входной массив коэффициентов промежуточных переменных,  $(N-1) \times (N-1)$  ( $L+1$ );

IH — входной массив вторых индексов промежуточных переменных;

N — число рабочих аргументов плюс единица;

T — номер последнего ряда селекции;

Метод. По алгоритмам МГУА получаются в виде набора промежуточных переменных (пирамиды решения) [13]. Уравнение в развернутом виде можно получить, исключив эти переменные.

Предлагаемая подпрограмма исключает промежуточные переменные последовательно в каждой из моделей предпоследнего ряда. Подпрограмма может работать только при  $L = 3$ , т. е. для линейных уравнений.

#### ПРОГРАММА 20

```

1.  'PROCEDURE'КОЭФ(IH,N,T,IH,XS,,'
2.  ,ARRAY'KH,XS,,INTEGER'ARRAY'IH,,INTEGER'N,T,,'
3.  ,BEGIN'ARRAY'K/C/1:N,1:N/,,INTEGER'I,J,H,G,F,R,,REAL'DM,,'
4.  ,FOR'H:=1'STEP'1'UNTIL'N-I'DO'BEGIN'K(/1,1/):=KH(/T-1,H,1/),,,'
5.  ,FOR'I:=2'STEP'1'UNTIL'N'DO'R/K(1,I/):=0..R:=TH(/T-1,H,I/)+1..
6.  ,F:=H+1..R/C(1,F/):=KH(/T-1,H,2/),,R/C(1,R/):=KH(/T-1,H,3/),,,'
7.  ,FOR'I:=T-2'STEP'-1'UNTIL'1'DO'BEGIN'FOR J:=2'STEP'1'
8.  ,UNTIL'N'DO'FOR G:=1'STEP'1'UNTIL'N'DO'K(/J,G/):=0,,'
9.  ,FOR'J:=2'STEP'1'UNTIL'N'DO'IF'K(/1,J/)'NE'0'THEN'
10. ,BEGIN'F:=J,,R:=IH(/1,J-1/)+1..K(/J,1/):=KH(/1,J-1,1/),,,'

```

```

17. ,FOR'I:=1'STEP'1'UNTIL'J-1'DO'BEGIN'FOR H:=J-1'STEP'-1'
18. ,UNTIL'1'DO'A(/H,J-1/):=A(/H,J-1/)-('IF'N'EQ'J-T'THEN'
19. ,A(/J,N)*A(/J-1,J/)),'ELSE'-(A(/J,N)*A(/J-1,J/))'END'',
20. A(/J,J/):=1/A1(/J,1/),,'IF'J'GE'3'AND'M'EQ'0'THEN'BEGIN'
21. 'IF'D:=0.,'FOR H:=K-1'STEP'1'UNTIL'KK'DO'BEGIN'W:=0.,
22. 'FOR'I:=1'STEP'1'UNTIL'J'DO'W:=W+A(/L+1,I/)*K1(/I,H/),,,'
23. 'Phi:=Phi+W*X1(/L+1,H/),,'D:=D*W**2'END'','Phi:=Phi/SQRT(D*Phi),
24. 'IF'Phi=D1'GE'0.001'THEN'BEGIN'Phi:=Phi,,G:=J'END'','END'',
25. 'IF'J'EQ'M'THEN'GOTO'C,,'GOTO'B,,'
26. E:'FOR'I:=1'STEP'1'UNTIL'L+1'DO'A(/I,J/):=0..B:'END'',
27. C:'IF'W'GT'1'THEN'BEGIN'FOR I:=1'STEP'1'UNTIL'M'DO'
28. A2(/I/):=A(/L+1,I/),,A2(/L+1/):=A1(/L+1,I/)'END'ELSE'M:=C
29. ,END,,EOP'

```

11.  $K(J,J) := K(J,J) * K(J,J) * K(J,J) * K(J,J)$

12.  $K(J,J-1,3) := K(J,J-1,3) * K(J,J-1,3)$

13.  $BEGIN' IF J < N THEN K(J,J) := 0, FOR G := 2 STEP 1$

14.  $UNTIL' N DO' K(J,J) := K(J,J) + K(G,J)$

15.  $PRINT' ((I,1,12,12,12,12,12,12,12,12,12,12))$

16.  $DO' (J=1, K(J,J)) , PRINT' () END' END' EOP'$

#### ПОДПРОГРАММА YU

Составление условных уравнений  $i, j$ -й пары по заданному виду частного описания (прогр. 21). Выходной массив  $X_1$  является входным для процедуры HY.

*Обращение.* УУ ( $X_1, X_P, I, J, L, N, NI$ ).

*Описание параметров:*

$X_P$  — входной массив аргументов размерности  $N$  на  $NI$ ;

$X_1$  — выходной массив условных уравнений для  $j, i$ -й пары размерности  $(L+1)$  на  $NI$ ;

$I, J$  — индексы аргументов. Указывают строки массива  $X_P$ , из которых будут сформированы условия уравнения,

$L$  — длина частного описания;

$N$  — число аргументов плюс единица;

$NI$  — число точек наблюдения.

*Требование функции и подпрограммы.* Нет. Метод. Для  $i, j$ -й пары аргументов матрица условных уравнений заданного вида составляется с помощью перезаписи соответствующих частей массива  $X_P$ .

#### ПРОГРАММА 21

```
1. PROCEDURE YU(X1,XP,I,J,L,N,NI).
2.   INTEGER'I,J,L,N,NI,'ARRAY X1,XP,, 'BEGIN' INTEGER'H,F,,'
3.   'FOR'H:=1 STEP 1 UNTIL'N DO' 'BEGIN' F:=3,
4.     X1((1,J)):=1,X1((2,H)):=XP((1,H)),X1((3,H)):=XP((J,H)),
5.     'IF'L>3'THEN 'BEGIN' F:=F+1,
6.       X1((F,H)):=XP((1,H))*XP((J,H)) END' ,
7.     'IF'L>4'THEN 'BEGIN' F:=F+1,X1((F,H)):=XP((1,H))*2' END' ,
8.     'IF'L>5'THEN 'BEGIN' F:=F+1,X1((F,H)):=XP((J,H))*2' END' ,
9.     F:=F+1,X1((F,H)):=XP((N,H)) END' END' EOP'
```

#### ПОДПРОГРАММА HY

Составление матрицы нормальных уравнений из указанной части матрицы условных уравнений (прогр. 22).

*Обращение.* HY ( $S, X_1, C, K, L$ ).

*Описание параметров:*

$X_1$  — входной массив условных уравнений размерности  $(L+1)$  на  $NI$ ;

$S$  — выходной массив нормальных уравнений размерности  $(L+1)$  на  $(L+1)$ ;  $C, K$  — номера начального и конечного столбцов той части массива  $X_1$ , из которой формируется матрица нормальных уравнений ( $K > C$ );

$L$  — длина частного описания.

Метод. Матрица нормальных уравнений составляется в соответствии с методом наименьших квадратов.

ПРОГРАММА 22

1. 'PROCEDURE HY(S,X1,C,K,L),, 'INTEGER'C,K,L,, 'ARRAY'S,X1,,

2. 'BEGIN' 'INTEGER'M,G,F,, 'FOR' M:=1 STEP 1 UNTIL'L+1 DO'

3. 'FOR' G:=1 STEP 1 UNTIL'L+1 DO'S((H,G)):=0,

4. 'FOR' H:=1 STEP 1 UNTIL'L+1 DO' 'FOR' G:=1 STEP 1 UNTIL'L+1 DO'

5. 'FOR' F:=0 STEP 1 UNTIL'K DO'S((G,W)):=S((G,W))+

6. X1((H,F)\*X1((G,F)) END' EOP'

#### ПОДПРОГРАММА Y

Вычисление оценок промежуточных переменных в конце каждого ряда селекции и заполнение соответствующего массива (прогр. 23).

*Обращение.* У ( $L, A, N, X, A2, X1$ )

*Описание параметров:*

$X_1$  — выходной массив аргументов размерности  $(L+1)$  на  $NI$ ;

$A2$  — выходной вектор коэффициентов промежуточных переменных длиной  $(L+1)$ ;

$X$  — выходной массив значений промежуточной переменной размерности  $N$  на  $NI$ ;

$L$  — число аргументов промежуточной переменной;

$A$  — номер промежуточной переменной (строка массива  $X$ , куда будут записаны оценки промежуточной переменной);

$N$  — число аргументов плюс единица.

*Требование подпрограммы и функции.* Нет.

Метод. Оценки значений промежуточных переменных находятся по формуле

$$X_{A,h} = \hat{q}_h = X \sum_{g=1}^L A_{2,g} X_1_{g,h}$$

#### ПРОГРАММА 23

```
1. 'PROCEDURE Y(L,A,N,X,A2,X1),, 'INTEGER'L,A,N,, 'ARRAY'X,A2,X1,,'
2. 'BEGIN' 'INTEGER'I,J,, 'REAL'S,, 'FOR' I:=1 STEP 1 UNTIL'L DO'
3.   'BEGIN' S:=0, 'FOR' J:=1 STEP 1 UNTIL'L DO'
4.     S:=S+2(X(J)*X1((I,J)),X((I,1)):=S'END' END' EOP'
```

#### ПОДПРОГРАММА GRAF

Построение графика промежуточной переменной, имеющей минимальную величину среди квадратичной ошибки (СКО) на обучательной, проверочной и экзаменационной последовательностях (прогр. 24).

*Обращение.* GRAF ( $X, N, NI, I, K2, FM, DPhi, T$ )

*Описание параметров:*

$X$  — массив отсчетов промежуточных переменных последовательного ряда,  $N$  на  $NI$ ;

$I$  — вектор ранжированных номеров точек наблюдения длиной  $NI$ ;

$K2$  — общая длина обучающей и проверочной последовательностей;

$FM$  — минимальное значение выходной величины;

$DPhi$  — размах выходной величины ( $DPhi = \Phi_{max} - \Phi_{min}$ );

$N$  — число аргументов плюс единица;

$NI$  — число точек наблюдения;

$T$  — длина интервала упреждения.

*Требование подпрограммы и функции.* SQRT.

**Метод.** Из всех моделей предпоследнего ряда выбирается та, у которой суммарная величина среднеквадратичной ошибки на обучающей, проверочной и экспериментальной последовательностях минимальна. Строится график выходной величины  $\Phi$  и ее оценки  $\hat{\Phi}$ . Знаками  $+$  или  $-$  обозначаются, выше или ниже выходной величины находятся ее оценки, а также печатаются численные значения отклонений оценок выходной величины от ее истинного значения. По абсциссе графика откладываются номера точек наблюдений в естественном порядке следования, по ординате — значения функции.

Перед построением графика на печать выводятся: номер модели, величина СКО, цена десетни ординаты.

```

1.  PROCEDURE GRAF(Y,N,N1,I,K2,QM,DY,TY),
2.  ,ARRAY X,, INTEGER,ARRAY I,, INTEGER N,N1,K2,TY,,REAL'QM,DY,,,
3.  'BEGIN' INTEGER I,J,A,B,IK,II,,ARRAY Y1(0:101),,STRING'P2,,,
4.  'REAL'D,MAX,MN,DK,, STRING,ARRAY P1(0:101),,STRING'P2,,,
5.  A:=N-1,II:=N1-TY,,FOR I:=1,STEP 1,UNTIL A'DO,,BEGIN D=0,,,
6.  'FOR J:=1,STEP 1,UNTIL II'DO,D:=(X(I,J)-X(N,J))*2,,,
7.  'IF I EQ 1 OR D LT DK THEN' BEGIN DK:=D,,IK:=I END'
8.  'END',,DK:=SORT(DK/11)*10,,MAX:=MIN:=X(IK,1),,H:=0,,,
9.  'FOR I:=IK,N'DO,,BEGIN H:=H+1,,FOR J=1,STEP 1,UNTIL N1'DO'
10. 'BEGIN X1:=H,I,J)/:=X(I,J)/,,IF'MAX LT X(I,J)''THEN'
11. MAX:=X(I,J)ELSE' IF'MIN GT X(I,J)''THEN MIN:=X(I,J)
12. 'END',,D:=(MAX-MIN)*0.01,,P2:=' ',,H:=N1-TY,,,
13. 'PRINT',(I+1) НОМЕР РЕШЕНИЯ,15,10X,19НСКО. ОБ.,НР. И ЭКЗ,,,
14. E18.5,//,13Н-ЦЕНА ДЕЛЕНИЯ,Е18.5,//3Х,11E10.3,2Х,
15. ИОНУЩЕНИЕ,/6Х,1Н+,9(X(H-),1Н,),9(1Н-),1Н+,/,/,
16. IK,DK,D*DK,'FOR I:=0,STEP 10,UNTIL 100'DO,DM+(MIN*D*I)*10,,,
17. 'FOR I:=1,STEP 1,UNTIL N1'DO,,BEGIN A:=(X1(2,I)-MIN)/D,,,
18. B:=' IF'I LEH' THEN'(X1(1,I)-MIN)/D ELSE'A,,,
19. 'FOR J:=0,STEP 1' UNTIL 101'DO,P1(J):=' ',,,
20. 'FOR J:=10,STEP 10' UNTIL 100'DO,P1(J):=' 077',,
21. P1(0):=' +',,IF'A>0'THEN'',BEGIN'
22. 'FOR J:=B STEP 1' UNTIL'A'DO,P1(J):=' -',,END'ELSE',
23. 'FOR J:=A+1,STEP 1' UNTIL'B'DO,P1(J):=' +',,,
24. DK:=(X1(2,I)-X1(1,I))*2,,P1(B):=' *',,,
25. 'IF'I LEH' THEN'P1(A):=' -',,IF'I LEH' THEN'
26. PRINT(NEWLINE,P2,P1,DK)'ELSE'PRINT(NEWLINE,P2,P1),,
27. 'IF'I EQ K2 THEN'P2:=' ',,IF'I EQ H THEN'P2:='
28. ',,PRINT',,IK:=V5,,IF'5*TK'EQ'Г' THEN'PRINT',(1Н+,6Х,
29. 100(1Н-))'END',,PRINT(NEWLINE),END'EOF'
```

#### ПРОГРАММА 24

**Используемые подпрограммы:** KOV, GRAPH2, ГЕНЕР, АГМ, ABS, ENTER.  
**Метод.** Синтез многомерной математической модели осуществляется в процессе неполного направляемого перебора различных комбинаций аргументов [14]. Сложность математических и динамических характеристик многомерных объектов более вероятном направлении. Получение математических моделей можно представить как циклический процесс, состоящий из следующих операций: генерация множества гипотез о пути усложнения математической модели; отбор по специальным правилам (с использованием принципов внешних дополнений) совокупности наиболее удачных решений; уточнение отобранных решений коррекции состава аргументов, организация прямой генерации гипотез следующего ряда. В зависимости от типа поставленной задачи математическое описание можно получить в одном из следующих видов:

для описания статики объектов

$$y(t) = f(\bar{x}(t)); \quad (3.12)$$

для описания динамики

$$y(t) = f(\bar{x}(t), \bar{x}(t-\tau), \bar{x}(t-2\tau), \dots, \bar{x}(t-k\tau)); \quad (3.13)$$

для краткосрочного прогнозирования

$$y(t+k\tau) = f(\bar{x}(t), \bar{x}(t-\tau), \bar{x}(t-2\tau), \dots, \bar{x}(t-k\tau)), \quad (3.14)$$

где  $\bar{x}(t)$  — вектор значений аргументов. В момент времени  $t$ ,  $y(t)$  — значение идентифицируемой характеристики в момент времени  $t$ ;  $\tau$  — шаг квантования учебного материала предыстории;  $k\tau$  — максимальная «глубина» учета предыстории;  $k_1$  — время упреждения в прогнозирующей модели.

На произвольном  $S$ -м ряду (шаге) математическая модель усложняется согласно выражению [14]

$$y_i^{(S)} = y_i^{(S-1)} + a_i \hat{z}_i,$$

где  $y_i^{(S)}$  — промежуточное решение  $S$ -го ряда;  $y_i^{(S-1)}$  — решение, выбранное на  $(S-1)$ -м ряду;

$$a_i = \frac{\sum_{j \in L_i} (\hat{\varphi}_j z_j)}{\sum_{j \in L_i} (\hat{z}_{j,i})^2}; \quad \hat{z}_i = z_i - b_i y_i^{(S-1)}; \quad (3.15)$$

$$b_i = \frac{\sum_{j \in L_i} y_j^{(S-1)} z_j}{\sum_{j \in L_i} (y_j^{(S-1)})^2}; \quad \hat{\varphi}_i = \varphi_i - \frac{1}{R} \sum_{j \in L_i} \varphi_j;$$

$\{\varphi_j\}$  — множество значений аппроксимируемой переменной;  $R$  — длина  $\{z_{j,i}\}$  — множество значений  $i$ -обобщенного аргумента  $S$ -го ряда;  $L_i$  — множество точек обучающей выборки. При  $S=1$   $y_i^{(S-1)}=0$ .

Множество обобщенных аргументов формируется по следующим правилам:

На первом ряду  $G_1 = \{x_i\} \cup \left\{ \frac{1}{x_i} \right\}$ , либо  $G_1 = \{x_i\}$  в зависимости от задания.

Из  $G_1$  выделяется  $F$  решений с минимальным значением критерия отбора — множество  $R_1$ .

На втором ряду множество обобщенных аргументов  $G_2 = G_1 \cup G_1 \otimes R_1$ .

либо, в зависимости от задания управляющего параметра,

$$G_S = G_1 \cup G_1 \otimes (R_1 \cup R_2 \cup \dots \cup R_{S-1}), \quad (3.16)$$

или  $G_S = G_1$ .  
При наличии в управляющих параметрах требования коррекции множество  $R_S$  дополняется множеством выбранных на  $S$ -м ряду обобщенных аргументов  $R'_S$ , затем из множества обобщенных аргументов отбирается  $G'_S$ , формирующееся

$$G'_S = R'_S \otimes G_S \quad (3.18)$$

и т. д. до

$$G^{(L)}_S = R^{(L)}_S \otimes G_S$$

Коррекции решений  $S$ -го ряда выполняются до тех пор, пока критерий отбора не перестает уменьшаться, либо пока число коррекций не достигнет максимально заданной величины.

На  $S$ -м ряду (с коррекциями или без них) выбирается  $\Gamma$  решений с минимальным значением критерия отбора ( $\Gamma > 1$ ). Окончательный выбор единственного решения  $S$ -го ряда выполняется в пользу того решения ряда, которое дает минимальное значение критерия отбора на  $S + 1$ -м ряду.

**Критерий отбора.** В программе применен комбинированный критерий отбора

$$\delta_{\text{от}} = TN(\text{OP} \cdot \delta_{\text{об}} + (1 - \text{OP}) \delta_{\text{пр}}) + (1 - TN) \left| \frac{\delta_{\text{об}} - \delta_{\text{пр}}}{\delta_{\text{об}} + \delta_{\text{пр}}} \right|, \quad (3.19)$$

где

$$\delta_{\text{об}} = \frac{\sum_{j \in L_1}^0 \Phi_j y_{S,j}}{\sum_{j \in L_1}^0 (\Phi_j)^2}, \quad \delta_{\text{пр}} = \frac{\sum_{j \in L_2}^0 \Phi_j y_{S,j}}{\sum_{j \in L_2}^0 (\Phi_j)^2}; \quad (3.20)$$

$\{y_{S,j}\}$  — множество решений  $S$ -го ряда;  $L_1$ ,  $L_2$  — множества точек обучающей и проверочной выборок;  $TN$  — параметр баланса между критерием точности и критерием несмещенности коэффициентов;  $\text{OP}$  — параметр баланса между оценками на обучающей и проверочной выборках в критерии точности.

Параметры  $\text{OP} (0 \leq \text{OP} \leq 1)$  и  $TN (0 \leq TN \leq 1)$  задаются пользователем в зависимости от типа решаемой задачи. При  $TN = 1$ ,  $\text{OP} = 0$  критерий (3.19) соответствует критерию регуляризации [5], при  $TN = 0$  — критерию несмещенности коэффициентов [14].

Приемы регуляризации. Первым  $M_1$  из  $M$  узлов интерполяции присваиваются случайные (точнее, псевдослучайные) номера от 1 до  $M_1$ . Экспериментальные точки со случайными номерами от 1 до  $R$  образуют обучающую выборку, а точки с номерами от  $R+1$  до  $M_1$  — проверочную. Этим достигается максимальное «перемешивание» точек для выражения характеристик обучающей и проверочной выборок. Кроме того, случайное перенесение позволяет в широких пределах изменять соотношение между количеством точек в обучающей и проверочной выборках.

**Критерий останова.** Усложнение математической модели в процессе самоограничения прекращается в двух случаях:

1. Число слагаемых в математической модели достигает перед заданного значения (значение идентификатора Ц4).

2. Выполняется условие

$$\delta_{\text{пр}}^{(S)} - \delta_{\text{пр}}^{(S+1)} < \varepsilon, \quad (3.21)$$

где  $\delta^{(S)}$  — определяется по формуле (3.20) для точек проверочной выборки;  $\varepsilon = M/(9 \cdot 10^8)$ ;  $M$  — общее число точек.

**Структура программы.** Программа состоит из главной программы для оригиналования и ввода исходных данных, подпрограммы АГМ (собственно алгоритма МГУД) и вспомогательных подпрограмм КОУ, ГЕНЕР, GRAPH2.

Основная часть алгоритма выражается в отдельной подпрограмме для удобства пользователя, который в своей программе может многократно обращаться к подпрограмме АГМ, изменяя исходные данные и параметры. Например, в общей программе можно получить аппроксимацию с использованием одной группы аргументов, а описание остатка — на основе другой группы аргументов.

#### Описание параметров:

$M$  — общее число экспериментальных точек;

$N$  — число исходных аргументов в массиве исходных данных;

$K$  — число обобщенных аргументов первого ряда селекции;

$R$  — длина учитываемой предистории, соответствующая величине  $k$  в соотношении (3.13) и (3.14);

$M_1$  — длина учитываемых исходных данных (3.13) и (3.14);

$\Gamma$  — глубина прогноза, соответствующая величине  $k_1$  в соотношении (3.14);

$\Delta_3$  — периодичность печати графика (по рядам селекции);

$\Delta_4$  — ограничение на максимально возможное число рядов селекции;

$\Delta_5$  — ширина графика;

$\Delta_7$  — параметр формы селекции;

$W$  — максимально возможное число коррекций на каждом ряду;

$\Delta_9$  — задающее число генератора псевдослучайных чисел;

$TN$  — вес критерия точности;

$\Gamma$  — параметр свободы выбора решений.

**Последовательность работы программы.** Вводятся управляющие параметры, формируются массивы исходных данных, осуществляется контрольный вывод значений параметров и элементов массива исходных данных и обращение к собственно алгоритму самоорганизации — подпрограмме АГМ.

**Примечания:** 1. При  $K = 2N$  — с положительными и отрицательными степенями, при  $K = 2N$  — с положительными и отрицательными степенями, при  $K = 2N$  — с положительными и отрицательными.

2. Число точек проверочной последовательности равно разности  $N_1 - R$ .  $M = M_1$  последних экспериментальных точек образуют отдельную контрольную выборку, не участвующую ни в обучении коэффициентов, ни в выборе аргументов.

Параметр формы селекции  $\Delta_7$  разлагается в программе на три составляющие:  $R_6$ ,  $R_7$  и  $R_8$ , где  $\Delta_7 = R_6 \cdot 10 + R_7 \cdot 10 + R_8$ . При  $R_6 = 0$  предлагаются аргументы на произвольном  $S$ -м ряду образуются в соответствии с выражением (3.16), а при  $R_6 = 1$  — с (3.17). Аналогично, при  $R_7 = 0$  коррекция осуществляется на множестве, сформированном в соответствии с выражениями (3.16) и (3.18), а при  $R_7 = 1$  — в соответствии с (3.17) и (3.18). Параметр  $R_8$  определяет, используется или нет принцип неокончательности решений с увеличением числа слагаемых. При  $R_8 \neq 1$  окончательный выбор аргументов модели  $S$ -го ряда осуществляется из  $\Gamma$  претендентов на  $S + 1$ -м ряду, а при  $R_8 = 1$  — сразу же после выполнения коррекций  $S$ -го ряда.

После ввода величин  $M$ ,  $N$ , ...,  $TN$  определяются размеры исходного массива аргументов, массива аппроксимируемой переменной и аппроксимации:  $\Omega_B [1 : M]$ ,  $1 : N]$  — массив аргументов;  $\Phi [1 : M]$  — массив аппроксимируемой переменной;  $Z [1 : M]$  — массив аппроксимаций (решений).



```

11. 'PRINT'', (12ЧИСЛО ТОЧЕК=,13)'',M., 
12. 'PRINT'', (20ЧИСЛО ВХ.ПЕРЕМЕННЫХ=,13)'',N., 
13. 'PRINT'', (21ЧИСЛО ОБОРЛ. ПЕРЕМ-=,13)'',E., 
14. 'PRINT'', (17ЦДЛНА ОБ.ВЫБОРГ=,13)'',R., 
15. 'PRINT'', (16ЧИСЛО РАВ.ТОЧЕК=,13)'',M1., 
16. 'PRINT'', (ЗОИДЛНА УЧИТАВАЕМОЙ ПРЕДЕСТОРИ=,13)'',M1., 
17. 'PRINT'', (17ЦДЛЖЕНИЯ ПРОГНОЗА=,13)'',Ц2., 
18. 'PRINT'', (29ЧИСЛО ИНТЕРВАЛЬНОСТЬ ПЕЧАТИ ГРАФИКА=,13)'',Ц3., 
19. 'PRINT'', (21ЦПОСТАНОВКА СЧЕТА ПОСЛЕ,Т4,2Х,4ПРЯДА)'',Ц4., 
20. 'PRINT'', (14ЧИСЛО ГРАФИКА=,13)'',Ц5., 
21. 'PRINT'', (14ЧИСЛО СЛЕЖЕНИИ=,14)'',Ц7., 
22. 'PRINT'', (23ЧИСЛОВОБОДА ВЫБОРА РЕШЕНИЙ=,13)'',Г., 
23. 'PRINT'', (4ЧИСЛО МОРРЕКИИ=,13)'',Н., 
24. 'PRINT'', (23ЧИСЛО ЧИСЛО ГЕЛ.СЛ.ЧИСЕЛ=,13)'',Ю., 
25. 'PRINT'', (17ЧИСЛО ОБЩ.ВЫБОРИГ=,FB_4)'',ОР., 
26. 'PRINT'', (29ЧИСЛО ВЫП. ТОЧНОСТИ=,FB_4)'',TH., 
27. MIN:=M_0.90*09., 'FOR'J:=1'STEP'1'UNTIL'M1'DO'K(J/):=0..M:= 
28. M-Ц1., 'IF'M1'GT'N-Ц2'THEN'M1:=M-Ц2., 'IF'R'GT'N'THEN'BEGIN' 
29. 'FOR'I:=1'STEP'1'UNTIL'N'DO'FOR'J:=1'STEP'1'UNTIL'M1'DO', 
30. 'BEGIN'OB(J,I/):=OB(J,I/)+0.01., 'IF'ABS(OB(J,I/))LT'MIN 
31. 'THEN'OB(J,I/):=MIN'END' END'., 'FOR'J:=1'STEP'1'UNTIL'M-Ц2 
32. 'DO'Ф(J/):=Ф(J+Ц1+Ц2/). R6:=ENTER(Ц7/100).,R7:=ENTER 
33. ((Ц7-Р6*100)/10).,R8:=Ц7-Р6*100-Р7*10., 'FOR'J:=1'STEP'1 
34. 'UNTIL'M1'DO', BEGIN'НЕР(Ю,Б)., DR(J/):=Р'END'., 
35. 'FOR'J:=1'STEP'1'UNTIL'M1'DO', BEGIN'И(J/):=J..C:=DR(J/)., 
36. 'FOR'Т:=1'STEP'1'UNTIL'M1'DO', IF'C'LT'DR(T/)'THEN'BEGIN' 
37. C:=DR(T/)., H(J/):=T'END'., DR(H(J/)):=0'END'., 'PRINT'', 
38. (/,9ННОМ.ТОЧЕК,1218)', 'FOR'J:=1'STEP'1'UNTIL'M1'DO'M(J/)., 
39. WФ:=Ц01:=0., 'FOR'J:=1'STEP'1'UNTIL'R'DO'WФ:=WФ*G(M(J/))/R., 
40. 'FOR'J:=1'STEP'1'UNTIL'M1'DO', BEGIN'Ф(J/):=Ф(J/)-WФ., 
41. X1(J/1,J/):=1..Y(J/):=Y3(J/):=MIN*2'END'., JФ:=0., 
42. 'FOR'J:=1'STEP'1'UNTIL'R'DO'DФ:=DФ+Ф(M(J/))/**2., 'FOR'J:= 
43. R+1'STEP'1'UNTIL'M1'DO'ДФ:=ДФ+Ф(M(J/))/**2., R1:=1/МН., 
44. B:=WФ/R., C:=WФ/(M1-R)., 'PRINT'',(/,15MAT.ОКНАДИЕ Ф=,, 
45. M16.6,5Х12ИДИСЛЕРСКИЙ Ф=,2,(E16.6,5X)'',MФ,B,C,P:=1.,Q1:=0.,

46. Ц1:=1..,0ИФ:=1..,NET:0И1:=1/АИН., 'FOR'12:=1'STEP'1'UNTIL'R'DO', 
47. S(/T2+1/):=1/AИН.,S(/1/):=-1/AИН.,I:=1., 'IF'R6=0'THEN'J:=Р., 
48. 'FOR'1:=1'STEP'1'UNTIL'1ДО', 'FOR'11:=1'STEP'1'UNTIL'R'DO', 
49. 'FOR'J1:=1'STEP'1'UNTIL'1Д1+1ДО', 'BEGIN'FOR'J:=1'STEP'1 
50. 'UNTIL'JДДО', 'IF'11'LE'R'THEN'Л(J/):=ДЕ(J+J1-1,11/)*Д(J,J/), 
51. 'ELSE'Л(J/):=Х1(J,J/)/ОБ(J+J1-1,11-N/).,C:=0Р.,OCT:=ИН., 
52. K0V(R,M,M1,B,C,OCT,MIN,Y,Y1,Ф,L,H),J2:=Ц1..,Ц4: 
53. 'IF'OCT'LT'S(/J2/)'THEN'BEGIN'IF'(OCT'GE'S(/J2-1/))'THEN' 
54. 'BEGIN'FOR'J:=T+1'STEP'-1'UNTIL'J2+1'DO'S(J/):=S(/J-1/), 
55. 'FOR'J:=T'STEP'-1'UNTIL'J2'DO', 'FOR'0Z:=1'STEP'1'UNTIL'3'DO' 
56. Н1(/J,J/):=Н1(/J2,J-1/),S(/J2/):=ОCT..,Н1(/1,J2-1/):=11., 
57. Н1(/J,J2-1/):=T.,Н1(/3,J2-1/):=J1., 'GOTO'М45'END'.,J2:=J2-1., 
58. 'GOTOM'Н14'END'.,Н15:'END'., 'PRINT'',(/,4ННРД ,J2,10X, 
59. 16ЧИСЛО ИНСАИНЕ,10ХИКОРРЕВУН,5Х16ИЗИАЧ.КРНТ.ОТБОРА)', 
60. Р., 'PRINT'',(/,1VX,2ННХ,4Х,2ННZ,4Х,2ННу,9Х,2НХ,3Х, 
61. 2НХу,10Х,3НХСТ)', 'FOR'12:=1'STEP'1'UNTIL'R'DO', 'BEGIN' 
62. Н1:=Н1(/1,J2/).T:=Н1(/2,J2/).,J1:=Н1(/3,J2/)., 'FOR'J:=1 
63. 'STEP'1'UNTIL'M'DO'Y(J/):=Y5(J/)., 'FOR'J:=1'STEP'1'UNTIL' 
64. M'DC', 'IF'11'LE'R'THEN'Л(J/):=ОБ(J+J1-1,11/)*К1(T,J/), 
65. 'ELSE'Л(J/):=Х1(T,J/)/ОБ(J+J1-1,11-N/).,C:=0Р.,OCT:=ИН., 
66. K0V(R,M,M1,B,C,OCT,MIN,Y,Y1,Ф,L,H),S1(/1,J2/):=C., 
67. S1(/2,J2/):=B.,БМ:=ОCT.,I4:=Ц1+1-Ц1., 'IF'11'GT'N'THEN'11:=N- 
68. T1., 'PRINT'',(/,16Х,13,4Х,12,4Х,12,25Х,E16,9)',11,14,T,OCT., 
69. 'FOR'J:=1'STEP'1'UNTIL'M'DO', BEGIN'DR(J/):=Л7(J/):=Л(J/),, 
70. Л3(J/):=Y1(J/)'END'., 'FOR'0:=1'STEP'1'UNTIL'W'DO', 'BEGIN' 
71. ОХ:=ОY:=ОZ:=0.,I:=1., 'IF'R7=0'THEN'Т:=Р., 'FOR'Т:=1'STEP'1 
72. 'UNTIL'1'DO', 'FOR'J1:=1'STEP'1'UNTIL'Ц1+1'DO', 'FOR'11:=1'STEP'1 
73. 'UNTIL'K'DO', 'BEGIN'FOR'J:=1'STEP'1'UNTIL'M'DO', 'BEGIN' 
74. 'IF'11'LEN'ТHEN'Л(J/):=ОБ(J+J1-1,11/)*Х1(T,J/)'ELSE' 
75. Л(J/):=Х1(T,J/)/ОБ(J+J1-1,11-N/).,Л(J/):=Л(J/)*Л3(J/)'= 
76. 'END',C:=0Р.,OCT:=ИН.,K0V(R,M,M1,B,C,OCT,MIN,Y,Y1,Ф,L,H), 
77. 'IF'0CT'LT'EM'THEN', 'BEGIN'S1(/1,J2/):=C.,S1(/2,J2/):=B., 
78. EM:=0CT.., 'FOR'J:=1'STEP'1'UNTIL'M'DO', BEGIN'L3(J/):= 
79. Y1(J/),DR(J/):=Л(J/)'END'.,ОХ:=11.,ОY:=Т.,ОZ:=Ц1+1-Ц1., 
80. 'IF'11'GT'N'THEN'ОХ:=N-11'END'.'END',.

```

```

81. PRINT,,(39%,13,3X,12,3X,12,9%,E16.9)**,0%,0Z,0Y,IM,,          116. 'СВЧЗ'ТЕН' BEGIN'01:=0., 'FOR'J:=M-12'STEP'1'UNTIL'M'DO'
82. 'FOR'J:=1'STEP'1'UNTIL'M'DO'LT(J/):=DR(J/)'END'.,           117. $ $(J/)$ :=0.,GRAPH2(M,15,Z7,L), 'PRINT'',(14)МАССИВ РЕШЕНИЙ,
83. 'FOR'J:=1'STEP'1'UNTIL'M'DO'BEGIN'Y(J/):=L3(J/)',,           118. /,8E14.5)',,27'END'., 'ГРУПП'Ц4'ТЕН' 'СОТОМЕТ'.,
84. L3(J/):=X1(P+1,J/):=DR(J/)'END'., 'IF'R8=1'THEN' 'BEGIN'           119. 'FOR'J:=1'STEP'1'UNTIL'M'DO'$ $(J/)$ :=Y(J/)+M0'END' 'EOF'
85. 'IF'JM'LT'0M1'THEN' 'BEGIN' A(/P/):=S1(1,12/),,D(/P/):=           Генерация псевдослучайных чисел, равномерно распределенных в интервале
86. S1(1/2,T2/),,NUK:=M1(1,12/),,NUY:=M1(2,12/),,NUZ:=M1(3,12/)          (0,1) (прогр. 27).
87. ,, 'FOR'J:=1'STEP'1'UNTIL'M'DO'BEGIN'Y(J/):=Y(J/),,           Обращение: ГЕНЕР (I, CL).
88. L2(J/):=L3(J/)'END'., OM1:=JM'END'., 'GOTO'M19., 'END'.,           Описание параметров:
89. I:=1., 'IF'R6=0'THEN'I:=P., 'FOR'T:=1'STEP'1'UNTIL'I+1'DO'           I — заложенное число генератора псевдослучайных чисел (целое, нечетное),
90. 'FOR'J1:=1'STEP'1'UNTIL'CL+1,D0'FOR'J1:=1'STEP'1'UNTIL'R'DO'           CL — псевдослучайное число, получаемое из подпрограммы.
91. 'BEGIN' 'FOR'J:=1'STEP'1'UNTIL'M'DO', IF'11'LEH'THEN'           Требование подпрограммы и функции ENTIER.
92. L(J/):=05K(J+J1-1,11/)*X1(T,J/)'ELSE'L(J/):=X1(T,J/)/           Метод. Алгоритм получения псевдослучайных чисел аналогичен алгоритму,
93. 05(/J+J1-1,11-H/),,C:=0P., OCT:=TH., K0V(R,M,M1,B,C,OCST,MIN,Y,       используемому в сервисной библиотеке мониторной системы АЛГОЛ БЭСМ-6
94. Y1,Ф,L,W),, 'IF'OM1'GT'0CT'THEN' BEGIN A(/P/):=S1(1,12/),,           и основан на поразрядном перемешивании массива заложенного числа (I)
95. L(/P/):=S1(2,12/),,NUY:=M1(2,12/),,NUZ:=           ПРОГРАММА 27
96. M1(3,12/),,OM1:=OCT., 'FOR'J:=1'STEP'1'UNTIL'M'DO' 'BEGIN',
97. Y2(J/):=Y(J/),,L2(J/):=L3(J/)'END' 'END'., M19:'END'.,           1. 'PROCEDURE ГЕНЕР(YL,CL),,' INTEGER'YL., 'REAL'CL., 'BEGIN'
98. 'FOR'J:=1'STEP'1'UNTIL'M'DO' 'BEGIN'Y(J/):=Y2(J/),,Y3(J/):=           2. YL:=3125*YL., YL:=YL-(ENTIER(YL/67108864))*67108864.,
99. Y(J/),,X1(J+1,J/):=L2(J/)'END'., C1:=0., 'FOR'J:=R+1'STEP'1           3. CL:=YL/67108864'END' 'EOF'
100. 'UNTIL'M1'DO'C1:=C1+(Ф(J(J/))-Y(J(K(J/)))*2)/M1.,           ПОДПРОГРАММА KOV
101. OM1:=0., 'FOR'J:=1'STEP'1'UNTIL'R'DO' OM1:=OM1+(Ф(K(J/))-           Ортогонализация обобщенного аргумента относительно решения предыдущего
102. Y(K(J/))/)*2)/D0., П:=0., 'FOR'J:=1'STEP'1'UNTIL'R'DO'Pi:=Pi+       ряда (3.15), вычисление коефициента  $a_i$ , получение решения текущего ряда,
103. L2(M(J/))//R., 'IF'P'GT'1'THEN' BEGIN'P'FOR T:=1'STEP'1'UNTIL'       вычисление величины критерия отбора (прогр. 28).
104. P-1'DO'A(T/):=A(T/)*(1-A(P/)*A(P/)),,CO:=CO*(1-A(P/)*           Обращение. K0V (R, M, M1, B, C, OCT, MIN, Y, Y1, Ф, L, И).
105. A(P/))A(P/)*Pi'ENDIF',ELSE'CO:=R(1/)*Pi., ПОСТ:=M0-C0., M1Z:=M1           Описание параметров:
106. +1*M1Z., 'IF'ПХ'GT'N'THEN'NUK:=I-NUK., 'PRINT',/,7НЫБРАНЫ,           R — длина обучающей выборки;
107. R-I'DO'A(T/):=A(T/)*(1-A(P/)*A(P/)),,CO:=CO*(1-A(P/)*           M — общая длина выборки исходных данных;
108. F11.8,5X,21НОМЕРКА АПРОКСИМАЦИИ=,F11.8,5X,15НОДЕНКА НА СЛЕД.,           M1 — длина рабочей выборки;
109. 4НРНД=,E16.8,//,4X,10НПОСТ.ЧЛЕН=,E16.6,//,8X,7НМАССИВ,           R — коэффициент  $b_i$  при ортогонализации  $z_i$  в выражении (3.15);
110. 1ЭНКОДДИМЕНТОВ,6E16.6)',,NUX,NUZ,NUY,0Ц,C4,OM1,ПОСТ,'FOR'           C — коэффициент  $a_i$  в выражении (3.15);
111. I1:=1'STEP'1'UNTIL'P'DO'A(I1/),, 'FOR'J:=1'STEP'1'UNTIL'M'DO'           MIN=M/9*10^8
112. 'BEGIN'Z(J/):=Y(J/)-M0.,L(J/):=Ф(J/)+M0'END'., 'IF'Ц2           Y, Y1 — решения предыдущего и текущего рядов;
113. 'GT'О'ПНЕУ'PRINT',/,7НПРОЦ3,6E16.6)',, 'FOR'J:=M1'STEP'1           Ф — аппроксимируемая переменная;
114. 'UNTIL'M'DO'Z(J/),,B12:=B11-C1.,B11:=C1., 'IF'B12'LE'MIN           L — вводимый обобщенный аргумент;
115. 'THEN' 'BEGIN'P:=I4.,Q1:=Ц3'END'.,P:=P+1.,Q1:=Q1+1., 'IF'Q1           И — случайные номера точек рабочей выборки.
           Требование подпрограммы и функции ABS
           Метод. По вводимым решению предыдущего ряда У и обобщенному аргументу
           L восстанавливается решение и вычисляется критерий отбора в соответствии с выражением (3.15).
           Для каждого элемента массива обобщенного аргумента L проверяется условие
            $MIN < |L_j| < -\frac{1}{MIN}$  (3.22)

```

Если условие (3.22) не выполняется,  $L_j$  ограничивается сверху и снизу, чтобы это условие выполнялось (строки 5—7). Затем центрируются элементы массива L (строки 7—8), вычисляется коефициент ортогонализации (3.15) (строки 8—11), где  $b_i$  соответствует идентификатор B. Массив L ортогонализуется относительно У (строка 11), вычисляется коефициент С (строки 13—14) и находится решение геометрического ряда Y1 (строки 14—15). Затем вычисляется значение критерия отбора OCT в соответствии с выражением (3.19) (строки 16—22).

116. 'СВЧЗ'ТЕН' BEGIN'01:=0., 'FOR'J:=M-12'STEP'1'UNTIL'M'DO'
117. \$ $(J/)$ :=0.,GRAPH2(M,15,Z7,L), 'PRINT'',(14)МАССИВ РЕШЕНИЙ,
118. /,8E14.5)',,27'END'., 'ГРУПП'Ц4'ТЕН' 'СОТОМЕТ'.,
119. 'FOR'J:=1'STEP'1'UNTIL'M'DO'\$ $(J/)$ :=Y(J/)+M0'END' 'EOF'

ПОДПРОГРАММА ГЕНЕР
(0,1) (прогр. 27).
Обращение: ГЕНЕР (I, CL).

Описание параметров:
I — заложенное число генератора псевдослучайных чисел (целое, нечетное),
CL — псевдослучайное число, получаемое из подпрограммы.

Требование подпрограммы и функции ENTIER.

Метод. Алгоритм получения псевдослучайных чисел аналогичен алгоритму, используемому в сервисной библиотеке мониторной системы АЛГОЛ БЭСМ-6 и основан на поразрядном перемешивании массива заложенного числа (I)

1. 'PROCEDURE ГЕНЕР(YL,CL),,' INTEGER'YL., 'REAL'CL., 'BEGIN
2. YL:=3125\*YL., YL:=YL-(ENTIER(YL/67108864))\*67108864.,
3. CL:=YL/67108864'END' 'EOF'

ПРОГРАММА 27
1. 'PROCEDURE ГЕНЕР(YL,CL),,' INTEGER'YL., 'REAL'CL., 'BEGIN
2. YL:=3125\*YL., YL:=YL-(ENTIER(YL/67108864))\*67108864.,
3. CL:=YL/67108864'END' 'EOF'

ПОДПРОГРАММА KOV
Ортогонализация обобщенного аргумента относительно решения предыдущего ряда (3.15), вычисление коефициента  $a_i$ , получение решения текущего ряда, вычисление величины критерия отбора (прогр. 28).

Обращение. K0V (R, M, M1, B, C, OCT, MIN, Y, Y1, Ф, L, И).

Описание параметров:
R — общая длина обучающей выборки;
M — общая длина выборки исходных данных;
M1 — длина рабочей выборки;
R — коэффициент  $b_i$  при ортогонализации  $z_i$  в выражении (3.15);
C — коэффициент  $a_i$  в выражении (3.15);

OCT — значение критерия отбора;

MIN=M/9\*10^8 — малое число;
Y, Y1 — решения предыдущего и текущего рядов;
Ф — аппроксимируемая переменная;
L — вводимый обобщенный аргумент;
И — случайные номера точек рабочей выборки.

Требование подпрограммы и функции ABS
Метод. По вводимым решению предыдущего ряда У и обобщенному аргументу L восстанавливается решение и вычисляется критерий отбора в соответствии с выражением (3.15).

Для каждого элемента массива обобщенного аргумента L проверяется условие
 $MIN < |L_j| < -\frac{1}{MIN}$  (3.22)

## ПРОГРАММА 28

```

1. 'PROCEDURE'KOV(R,M,M1,B,C,OCT,MIN,Y1,Q,L,M), 'INTEGER'R,M,
2. M1., 'REAL'B,C,OCT,MIN., 'ARR'W,Y,Y1,Q,I., 'INTEGER',ARRAY'M.,
3. , 'BEGIN','INTEGER',F,TI., 'ARRAY'L1('1':W/..,'REAL',KUR,KOR,
4. DIS,DIS1,B,C3,C4:=C..,KOR:=KOR1:=DIS:=WTS1:=B:=C:=0.,
5. ,FOR,F:=1'STEP'1'UNTIL'M'DO,,BEGIN'',IF'ABS(L(/F/))>GT,1/MIN
6. , 'THEN'L(/F/):=1/MIN., 'IF'ABS(L(/F/))<MIN' THEN'L(/F/):=MIN
7. , 'END'.,B:=0., 'FOR'F:=1'STEP'1'UNTIL'R'DO,b:=B+L(/M(/F/))/R.,'
8. , 'FOR',P:=1'STEP'1'UNTIL'M'DO'L(/P/):=L(/F/)-b., 'FOR',F1:=1'STEP'
9. , 1'UNTIL'R'DO,, 'BEGIN',KOR:=KOR+L(/h(/F1/)))*Y(h(/F1/)),,DIS:=
10. DIS*Y(h(/F1/))**2'END'., 'IF'DIS>MIN'THEN'B:=KOR/DIS,,KOR
11. :=OCT., 'FOR',P:=1'STEP'1'UNTIL'M'DO,L(/F/):=L(/F/)-Y(h(/F/))*B.,
12. , 'FOR',F:=1'STEP'1'UNTIL'R'DO,, 'BEGIN',KOR:=KOR+L1(h(/F/))*/
13. *h(/h(/F/))),,DIS1:=DIS1+L1(h(/F/))**2'END'.,C:=777/9 +11.,
14. , 'IF'DIS1>CT' MIN'THEN C:=KOR/DIS1., 'FOR',F:=1'STEP'1'UNTIL'M
15. , 'DO,, BEGIN'Y1(F/):=Y(F/)*L1(/F/)*C..,L(/F/):=L(/F/)+E'END'.,
16. OCT:=C3:=0., 'FOR',F:=1'STEP'1'UNTIL'R'DO,C3:=C3+Q(h(/F/))*/
17. 2., 'FOR',F:=1'STEP'1'UNTIL'M'DO,CT:=UCT+((Q(h(/F/))-
18. Y1(h(/F/)))*2)/C3.,DIS:=C1.,OCT:=C3:=0.,
19. , 'FOR',F:=R+1'STEP'1'UNTIL'M1'DO,C3:=C3+Q(h(/F/))*/2.,
20. , 'FOR',F:=R+1'STEP'1'UNTIL'M'DO,OCT:=OCT+(Q(h(/F/))-
21. Y1(h(/F/)))*2)/C3.,OCT:=KOR*(C4*DIS*(F-C4)*OCT)+(1-KOR)*
22. ABS(C(OCT-DIS))/(OCT+DIS))'END'.'EOP'

```

## ПОДПРОГРАММА GRAPH2

Сервисная, предназначена для печати сравнительного графика аппроксимации переменной и аппроксимации (прогр. 29).

### Обращение. GRAPH2(M, Ц5, Z1, Z2)

#### Описание параметров:

M — длина сравниваемых массивов;

Ц1, Ц2 — сравниваемые массивы

#### Требуемые функции ENTIER.

*Последовательность работы.* Для обоих массивов Z1 и Z2 определяется общее минимальное (M1) и максимальное (M<sub>A</sub>) значение (строки 4—10).

При Ц5=100 строится график Z<sub>1</sub><sub>i</sub> = f(Z<sub>2</sub><sub>i</sub>). Если коэффициент корреляции между Z1 и Z2 равен единице, точки графика образуют прямую линию.

При Ц5 ≤ 100 на листе АЛПУ одновременно печатается значение Z<sub>1</sub><sub>i</sub> и Z<sub>2</sub><sub>i</sub> для всех точек массивов Z1 и Z2 (строки 17—24). Максимальная ширина графика равна 100 позициям при Ц5 = 100. Длина графика определяется величиной M.

В результате счета на АЛПУ выполняются: значения M1 и M<sub>A</sub> (строки 7—8); линия начала графика (ось абсцисс) (строка 8); символами ж — точки в поле корреляции (при Ц5 = 100) (строка 16); символами «\*» и «!» — соответственно значения Z<sub>1</sub><sub>i</sub> и Z<sub>2</sub><sub>i</sub> (строка 24); символом «Х» — близкое расположение или совпадение значений Z<sub>1</sub><sub>i</sub> и Z<sub>2</sub><sub>i</sub> (строка 22).

## ПРОГРАММА 29

```

1. 'PROCEDURE'GRAPH2(M,Ц5,Z1,Z2), 'INTEGER'M,Ц5,, 'ARRAY'Z1,Z2.,
2. , 'BEGIN','INTEGER',F3,F4,, 'STRING',ARRAY'JH('1:103/), 'REAL'
3. E,S1,M1,M., 'INTEGER',ARRAY'JL('1:2,1:M/),M1:=M4:=Z1(/1/),
4. 'FOR',F4:=1'STEP'1'UNTIL'M'DO,, 'BEGIN',IF'MT'GR'Z1(/F4/),THEN'
5. MT:=Z1(/F4/),, 'IF'MA'LE'Z1(/F4/)'THEN'MA:=Z1(/F4/),
6. , 'IF'MA'LE'Z2(/F4/)'THEN'MA:=Z2(/F4/),, 'IF'MI'GE'Z2(/F4/)'
7. , 'THEN'MT:=Z2(/F4/)'END'., 'PRINT',//10*3HM=F10.4,803HM=
8. F10.4)**,M1,M., 'PRINT',//1X1H0,10(10H*****X))''.
9. JH('1/):=JH('103/):='*,*, 'IF'Ц5>CT'10'THEN',BEGIN'Ц5:=Ц5-
10. 100.,E:=Ц5/(M-M1),,E1:=ENTIER(Ц5*0.64),,Ц5:=Ц5+100., 'FOR'
11. F4:=1'STEP'1'UNTIL'M'DO,, 'BEGIN',J('1,F4/):=ENTIER(Z1(/F4/)-
12. M1*E),J('2,F4/):=ENTIER(Z2(/F4/)-M1*E*0.64)'END'.,
13. , 'FOR',F3:=1'STEP'1'UNTIL'E1'DO,, 'BEGIN',FOR',F4:=2'STEP'1'UNTIL',
14. 102'DO,JH(/F4/):='*,*, 'FOR',F4:=1'STEP'1'UNTIL'M'DO,IF',
15. J('2,F4/)=F3-1'THEN',JH('1,F4/):='*,*, 'PRINT'(NEWLINE,
16. JH),END'.'END'.'ELSE', 'BEGIN',FOR',F3:=1'STEP'1'UNTIL'M'DO,'FOR',
17. F4:=1'STEP'1'UNTIL'2'DO,, 'BEGIN',J('1,F3/):=ENTIER(Ц5*(Z2(/F3/)-M1*
18. (Z1(/F3/)-M1)/(M-M1)),J('2,F3/):=ENTIER(Ц5*(Z2(/F3/)-M1)/
19. /(M-M1))'END'., 'FOR',F3:=1'STEP'1'UNTIL'M'DO,'BEGIN',FOR',
20. F4:=2'STEP'1'UNTIL'102'DO,JH(/F4/):='*,*, 'IF',J('1,F3/)=
21. J('2,F3/)'THEN',J('1,F3/):='*,*, 'X','ELSE', 'BEGIN'
22. J('1,F3/):='*,*, 'J('2,F3/):='*,*, 'X','END'., 'PRINT'(NEWLINE,
23. PRINT(NEWLINE,LI)'END'.'END'.'END'.'EOP'

```

## 4. КОМБИНАТОРНАЯ ПРОГРАММА для структурной идентификации объектов и процессов управления

Программа (прогр. 30) предназначена для построения моделей оптимальной сложности многометных объектов и процессов управления по данным наблюдений в классе заданных базисных функций, линейных по коэффициентам [36].

Язык программирования АЛГОЛ, ЭВМ БЭСМ-б.

Используемые подпрограммы. DES, MAX, NOS, PAR, GAUSS.

Метод. Предполагается, что зависимость (3.24) выходной величины от l входных аргументов является полиномиальной, причем наибольшая степень σ полинома задана. При этом структура и число N членов полинома определяются

однозначно:

$$N = \prod_{j=1}^l \frac{\sigma + j}{j} = \frac{(\sigma + l)!}{l!}; \quad (3.23)$$

где степени аргументов  $S_m$  принимают значения 0, 1, ...,  $\sigma$  и удовлетворяют ука- занному ограничению.

Задана определения модели оптимальной сложности решается в два этапа. На первом этапе из полного набора моделей различной сложности (общее их число  $2^N - 1$ ) отбираются по критериям регулярности или несмещенности  $F$ . Лучших структур, коэффициенты которых на втором этапе пересчитываются по всей вы- борке. При этом определяются: среднеквадратическая ошибка (СКО) аппроксимации  $\delta$  ( $A + B$ ) и СКО экстраполяции  $\delta$  ( $C$ ). Могут быть вычислены также значения комбинированных критериев типа  $p_{cm} + \delta$  ( $A + B$ ) или  $p_{cm} + \delta$  ( $C$ ).

Окончательный выбор лучшей модели выполняется самим исследователем или группой экспертов на основе критерия сгруппированной полученных  $F$  моделей, критерии регуляризации, среднеквадратических ошибок и (или) комбинированных критериев.

При идентификации объекта с несколькими выходами все описанные операции выполняются соответствующее число раз — для каждого выхода в отдельности. В программе используются две программы — собственно комбинаторный алгоритм, подпрограмма COMBI (прогр. 31) и программа-обращение (прогр. 30), где считываются исходные данные и выполняются необходимые подготовительные операции.

#### Описание параметров программы-обращения. Вводные:

CT — наибольшая степень аппроксимирующего полинома;

L — число аргументов;

N — число точек (длина массива) исходных данных;

МП, МЭ — длины проверочной и экзаменационной последовательностей

соответственно,

CB — число лучших моделей, отбираемых на 1-м этапе (свобода вы- бора);

KL — ключ, обеспечивающий синтез моделей только со свободным

членом ( $KL = 1$ ) либо без него ( $KL = 2$ ), при  $KL = 0$  выпол-

няется полный перебор;

CRI — задание критерия: регуляризость — при  $CRI = 0$ , смещение —

при  $CRI = 1$ ,  $p_{cm} + \delta$  ( $A + B$ ) — при  $CRI = 2$  и  $p_{cm} + \delta$  ( $C$ ) —

при  $CRI = 3$ ;

PRY — управление выводом на печать результатов аппроксимации

всех CB моделей: при 1 эти результаты печатаются, при 0 —

нет; нет, — не используется;

NM — число идентифицируемых уравнений (от тех же L аргументов);

X [1 : M, 1 : L] — массив исходных данных (аргументов);

Y [1 : M] — массив линий о выходной величине (функции) — считывается

в цикле NM раз.

#### Вычисления:

МО — длина массива обучающей последовательности A (МО =

= M — МП — МЭ);

N — число членов (коэффициентов) полной модели (включая сво-

бодный член);

PS [1 : N, 1 : CB] — массив для записи коэффициентов вычисленных моделей оптимальной сложности;

OS [1 : CB] — массив значений критерия отбора.

ПРОГРАММА 30

1. 'ALGOL:COMBI., 'BEGIN' INTEGER CT, L, M, MI, M3, CB, KL, CRI, PRY,
2. MM, F, I, J, N, MO,, READ(CT, L, M, MI, M3, CB, KL, CRI, PRY, NM), MO:=M-MI
3. -MO., 'PRINT',//('1X1БИАЛГОРИТМ COMBI/12X10(1H\*)'),,,
4. 'PRINT',//('5X20ЧИСЛО АРГУМЕТОВ L=13'),,L,, 'PRINT',(5X
5. 17MAXC.СТЕПЕНЬ CT=12),,CT,, 'PRINT',(5X15НРЕСТО ТОЧЕК M=
6. 14/5X11РАЗДЕЛЕНИЕ:5X3М0=13,5X3М1=13,5X3М2=13),,M,MO,MI,
7. MO., 'PRINT',//('5X19КРИТЕРИИ://4X1(1H\*)'),, 'IF'CRI=0'THEN'
8. 'PRINT',(17X12ПРЕДУЛНОСТЬ),, 'IF'CRI=1'THEN' 'PRINT',(17X
9. 1ЭННЕДОЧЕННОСТЬ),, 'IF'CRI=2'THEN' 'PRINT',(17X
10. 15НКОМПИЛДОВАННЫЙ),, 'IF'CRI=2'THEN' 'PRINT',(20X
11. 1Н(НЕСМ.+СКО)),'ELSE' 'PRINT',(20X18Н(НЕСМ.+Ш. НА ЭКЗ.)),,
12. 'END', 'PRINT',(5X19ИСВОЛА ВЫБОРА CB=13),,CB,, 'PRINT'
13. ,(5Х18ИЩЕСВОР),, 'IF'KL=0'THEN' 'PRINT',(15Х6ИДОЛЫЙ),,,
14. 'IF'KL=1'THEN' 'PRINT',(15X14НСО СВОБ.ЧЛЕНОМ),, 'IF'KL=2
15. 'THEN' 'PRINT',(15X14НЕЗ СВОБ.ЧЛЕНА),, 'IF'PRY=0'THEN'
16. 'PRINT',(5Х26ИДПРОКСИМАЦИЯ НЕ ВЫДОСИ),,N:=1,,FOR J:=1
17. 'STEP'1'UNTIL'L DO,N:=N\*(CT+J)/J,, 'BEGIN' ARRAY X<1:M,1:L>,
18. Y<1:M>,RS<1:N,1:CB>,OS<1:CB>,READ(X),, 'PRINT',(5X
19. 10АРГУМЕНТ:(10E12.3)),,X,, 'PRINT',(13X14НЧИМЕ МОДЕЛИ
20. /12X16(1H\*)),, 'FOR' I:=1'STEP'1'UNTIL'N DO,, 'BEGIN' READY,,
21. 'IF'NM'GT'1'THEN' 'PRINT',(7Х14ИЧИМЕ МОДЕЛИ),,I,,
22. 'PRINT',(7Х14ИЧИМЕ:(10E12.3)),,Y,,COMBI(CT,L,M,MI,M3,
23. CB,CRI,KL,PRY,X,Y,RS,OS),, 'FOR'F:=1'STEP'1'UNTIL'CB DO,
24. 'PRINT',(7ХБИМОДЕЛЬ3,5X1ИЧИМЕ КРИТЕРИЕЙ5,5/
25. (10E12.5)),,F,OS</F/),, 'FOR'J:=1'STEP'1'UNTIL'N DO,
26. RS<J,F>END,'END','END','EOP'

#### ПОДПРОГРАММА COMBI

Реализует комбинаторный алгоритм МГУА (см. прогр. 31).  
Обращение. COMBI (CT, L, M, MI, M3, CB, CRI, KL, PRY, X, Y, PS, OS).

Описание параметров. Вводимые: (см. с. 21)

В ч и с л я е м ы е:

ID [1 : M, 1 : N] — расширенный массив данных, построенный по данным

массива X с учетом степени CT (т. е. содержит все

степени и произведения исходных аргументов, а при

CT = 1 совпадает с X). Вычисляется в процедуре

FOD (см. ниже);

YS [1 : M] — массив результатов аппроксимации функции по каж-

дой из построенных CB моделей;

**FS, FS1 [1 : N, 1 : N+1]** — массивы систем нормальных уравнений, получаемых по методу наименьших квадратов (МНК) на частях А и В массива ИД. Стятся в процедуре **NOS**.

**P, P1 [1 : N]** — промежуточные массивы для записи коэффициентов частных моделей, получаемых на частях А и В;

**ОЭ [1 : CB]** — массив ошибок экстраполяции по части С исходных данных

**Требуемые функции и подпрограммы.** REGI, FOD, DES, NOS, MAX, PAR массива аргументов X и степени СТ автоматически формируются расширенный массив данных ИД, а также системы нормальных уравнений для полного полинома FS и FS1, если  $|CRI| \geq 1$ ; строки (20, 21). В блоке, содержащем строки (25—35), выполняется полный или усеченный (согласно значению KL) перебор частных моделей с отбором СВ наилучших по заданному критерию структур D [1 : N], после изменения состояния которого следует обращение к PAR. Отобранные структуры запоминаются как леситичные числа в массиве DF [1 : CB].

Вычисления второго этапа (пересчет коэффициентов) выполняются в блоке строк (43—49), где составляется новая нормальная система по данным массива ИД (от 1 до  $M_1 - M_2$ ), а структуры отобранных моделей восстанавливаются с помощью перевода каждого десятичного числа из DF в двоичное D [1 : N]. Здесь же вычисляются ошибки  $\delta(A + B)$  и  $\delta(C)$  для каждой модели. Если  $|CRI| \geq 2$ , то в конце подпрограммы вычисляются значения одного из комбинированных критериев.

Во время работы программы на печать выводятся: все вводимые (считываемые) числа и массивы с соответствующими комментариями; таблица степеней аргументов каждого члена полной модели под заглавием «Структура полного полинома»; число членов полной модели и число стравливаемых частных моделей; результаты аппроксимации, если  $PRY = 1$ ; лучшие модели и значения основного критерия.

## ПОДПРОГРАММА REGI

Получение результата добавления единицы к числу с произвольным основанием (см. прогр. 3).

**Обращение.** REGI (D, N, S).

**Описание параметров:**

D — число в виде одномерного массива (исходное и результативное);

N — размерность вектора D;

S — основание числа D

**Требуемые функции и подпрограммы.** Нет.

**Метод.** Единица суммируется с содержимым N-го разряда вектора D. Результат суммирования получается последовательным переходом единицы в высшие разряды числа при получении значения S в низшем разряде (который при этом обнуляется).

## ПОДПРОГРАММА FOD

Построение системы условных уравнений для полного полинома заданной степени от заданного числа переменных (прогр. 3).

**Обращение.** FOD (CT, L, M, X, XD).

**Описание параметров:**

СТ — степень полинома;

L — число аргументов;

M — число точек исходных данных;

X — двумерный массив исходных значений аргументов (размерность  $M \times L$ );

XD — расширенный массив данных размерности  $M \times N$ , где N — число членов полинома.

**Комментарий.** Параметр M является глобальным

**Требуемые функции и подпрограммы.** REGI.

**Метод.** Массив XD формируется по линейным массивам X в соответствии с формулой (3.24) с помощью постепенного обращения к REGI для получения массива П [1 : L] certain степеней, с которыми исходные аргументы входят в каждый член полного полинома. Для расшифровки структуры полинома получаемые массивы П печатаются в виде таблицы, в которой, например, при трех аргументах массивов X [1 : 3] строка вида 000 соответствует свободному члену, вида 100 — члену  $X_1$ , вида 021 — члену  $X_1^0 X_2^1 X_3^1 = X_2^2 X_3$  и т. д. Массивы коэффициентов P, P1, PS заполняются в соответствии с этой таблицей

## ПРОГРАММА 31

```

0.   *Л101*:DES,MAX,NOS,PAR..
1.   ,PROCEDURE COMBI(CT,L,M,M1,M2,CB,CRI,IL,PRY,X,Y,PS,OS),
2.   ,INTEGER CT,L,M,M1,M2,CB,CRI,IL,PRY,X,Y,PS,OS,
3.   ,DEC I,J,INTEGER C,F,I,J,N,O,CHK,,REAL,OL,R,Z,Phi,
4.   ,PROCEDURE REGI(D,N,S),,INTEGER N,S,,INTEGER'N,S,,'INTEGER'ARRAY'D,,,
5.   ,REGI',INTEGER'K,,K:=K+1,,M'R:=K-1,,IF'D/(K)'LT'S-1'THEN,
6.   DK/(K):=D/(K)+1'ELSE'BEGIN'D/(K):=0,,GOTO'M'ENDIF'END'',
7.   PROCEDURE FOD(CT,L,M,X,XD),,INTEGER CT,L,M,,ARRAY'X,XD,,,
8.   ,DEC IR'INTEGER'I,J,K,C,F,S,,INTEGER'ARRAY'II/(1:L)/,,'PRINT',
9.   ,(2БИСТРУКТУРА ПОЛНОГО ПОЛИНОМА),,'FOR'J:=1'STEP'1'UNTIL'
10.  L'DO'II/(J):=0..F:=0..,'FOR'K=0'STEP'1'UNTIL'CT' DO' BEGIN'
11.  II/(1):=K-1..II/(1):=0..M'REGI(II,L,K+1),S:=0..,'FOR'J:=1'STEP'
12.  1'UNTIL'L'DO'S:=S+II/(J)/,,'IF'S'NE'K'THEN'GOTO'M',F:=F+1.,
13.  ,PRINT',(5/2113)','FOR'J:=J'STEP'-1'UNTIL'1'DO'II/(J)/,,'FOR'
14.  I:=1'STEP'1'UNTIL'M'DO' BEG IN'XK/I,F/':=1..,'FOR'J:=1'STEP',
15.  1'UNTIL'L'DO' IF II/(L-J+1)/,'NE'0'THEN'XK/I,F/':=XD/(I,F)*
16.  K/(1,J)**II/(L-J+1/)'END'..,'IF'II/(1/)'LT'K'THEN'GOTO'K'END',
17.  ,END'FOD,,M:=1..,'FOR'J:=1'STEP'1'UNTIL'L'DO'N:=N*(CT+J)/J..,
18.  CK:=2*N-1..,'BEGIN'ARRAY'II/(1:M,1:N/),YS(1:M/),FS,
19.  FS1(1:N,1:N+1/),P,P1(1:N/),RS,OS(1:CB/),,INTEGER'LB,UB,
20.  ,INTEGER'ARRAY'D/(1:N/),DF/(1:CB/),,FOD(CT,L,M,X,XD),,NOS(N,
21.  1,M-MI-MO,LU,Y,FS),,'IF'CRI>0'THEN'NOS(N,M-MI-MO+1,M-MO,MD,
22.  Y,FS1),,LB='IF'KL=1'THEN'(CK+1)/2'ELSE'1..,UB:='IN'KL=2'THEN'
23.  (CK-1)/2'ELSE'CK..,CK:=UB-LB+1.,PRINT',(7X12НЧСЛО ЧЛЕНОВ,
24.  4H N=,13/13Х7МОДЕЛ4,110),,N,CK,,C:='IF'MI=0'THEN'1'ELSE',
25.  M-MI-MO+1.,'TF'KL=1'THEN'DES(LB-1,D,N,2)'ELSE'DES(0,D,N,2)..,
26.  ,FOR'Q:=LB STEP'1'UNTIL'UB'DO' BEG IN' REGI(D,N,2)..,
27.  PAP(D,N,FS,P),,'IF'CRI=0'THEN'BEG IN' QI:=0..,'FOR'J:=C'STEP'1

```

```

28. 'UNTIL'М-М0' BEGIN'Z:=0., 'FOR'I:=1'STEP'1'UNTIL'N'DO'
29. Z:=Z+P(У/1)*ID(У/J,1/), 'OR'I:=0+(Z-Y/J))*2'END'END'ELSE'
30. 'BEGIN'IN'PAR'D,N,FS4,P1), 'OR'I:=0., 'FOR'J:=1'STEP'1'UNTIL'М-М3
31. 'DO' 'BEGIN'Z:=0., 'FOR'I:=1'STEP'1'UNTIL'N'DO' 'BEGIN'
32. Z:=Z+P(У/1)*ID(У/J,1/), 'OR'I:=0+(1/1)*ID(У/J,1/)'END', 'OR'I:=0+
33. (Z-Ф)**2'END'END', 'IF'О'LT'Л*CB'THEN'F=0-LB-1'ELSE'
34. 'BEGIN'МАК(OS,CB,R,F), 'IF'О'LT'ГЕ'R'THEN'GOTO'Л'END', 'OS(F/)'
35. :=0..DF(F/R/):=0..Л.'END', 'Ф'0., 'FOR'J:=1'STEP'1'UNTIL'M'DO',
36. 'BEGIN'Ф'Ф'У(У/J)**2., 'IF'J=M-MC
37. 'THEN'Я:=Ф'END', 'Z'':=IF'CRI=0'AND'M'NE'0'THEN'Я-Z'ELSE'Я.,
38. 'FOR'I:=1'STEP'1'UNTIL'СР'DO'OS(У/I/):=OS(У/I)/Z., 'IF'CRI=0
39. 'THEN'PRINT', '(УХ12Н0М.НА'IPC0..)', 'ELSE'PRINT', '(УХ
40. 'ЯСЛЕДЕНИЕ:)', 'PRINT', '(10E12.5)', 'OS', 'IF'CRI'GE'2'THEN'
41. 'BEGIN'МАК(OS,CB,R,F), 'FOR'I:=1'STEP'1'UNTIL'СР'DO'RS(У/I/):=
42. OS(У/I)/R., 'PRINT', '(10Н0М.СМЕЛЬ/(10E12.5))', 'KS'END'
43. 'IF'РУY=1'THEN'PRINT', '(УХ14НАЧПРОКСМДИЯ:)', 'NO$N,1,
44. N-М3, ID, У, FS), 'FOR'F:=1'STEP'1'UNTIL'СВДО'BEGIN'
45. DES(UF(У/F),D,N,2), 'PAR(D,N,FS,F), 'FOR'I:=1'STEP'1'UNTIL'N'DO'
46. PS(У/I,F/):=F(У/I), 'OR'I:=0., 'FOR'J:=1'STEP'1'UNTIL'M'DO' 'BEGIN'
47. Z:=0., 'FOR'I:=1'STEP'1'UNTIL'N'DO'Z:=Z+P(У/I)*ID(У/J,1/),
48. 'IF'РУY=1'THEN'Y(У/J/):=Z., 'OR'I:=0+(Z-Y/J)**2., 'IF'J=M-M3
49. 'THEN'Я:=Z'END', 'OS(F/):=SORT(R/f), 'IF'M3'NE'0
50. 'THEN'ОС(У/F/):=SORT((ОУ-R)/(Ф-Я)), 'IF'РУY=1'THEN'PRINT'
51. ', '(УХ5К6Н0МДЕЛЬБ/(10E12.3))', F, Y'S'END', 'PRINT', '(УХ
52. 19НСКО ПОСЛЕ ПЕРЕСЧЕТА/(10E12.5))', 'OS', 'IF'M3'NE'0'THEN'
53. 'PRINT', '(УХ1ЮН0М.НА'ЭКЗ./(10E12.5))', 'OS', 'IF'CRI'GE'2'THEN'
54. 'BEGIN'IF'CRI=2'THEN'МАК(OS,CB,R,F)'ELSE'МАК(О3,CB,R,F),'
55. 'FOR'Р'':=1'STEP'1'UNTIL'СР'DO' 'BEGIN'О3(F/):='IF'CRI=2'THEN'
56. О3(F/)/R'EELSE'О3(F/)/R., 'OS(F/):=SORT(FS(F/)**2+О3(F/)**2)
57. 'END', 'IF'CRI=2'THEN'PRINT', '(УХ8Н0М.СК0)', 'ELSE'PRINT',
58. ', '(УХ12Н0М.О3.ЭКЗ.)', 'PRINT', '(10E12.5)', 'О3'END',
59. 'END', 'END'СМВ1'END',

```

**Примечания:** 1 В программе исходные данные на последовательности А, В и С разделяются не после предварительной ранжировки точек по дистерсии [14], а попаряд. Все вычисления ведутся с абсолютными запятыми данных, однако при необходимости пользователя может самостоятельно запрограммировать операции их нормировки и (или) ранжировки (в подпрограмме-обращении).

2. Госколько в программе выполняется полный перебор всех моделей различной сложности, ее возможности не беспределены. Установлено, что за привлекаемое время счета (приятия получаса на БЭСМ-6) решается задача с числом членов полинома модели  $N \leq 18$  при  $CRI = 0$  и  $N \leq 17$  при  $CRI \geq 1$ . Причем добавление одного члена увеличивает время счета примерно вдвое.

3. Данная программа пригодна для идентификации как статических, так и — при соответствующем представлении данных — динамических (разностных) объектов и процессов. Кроме того, пресобразовав должны образом данные массива  $X$ , можно перейти от поиска моделей в классе полиномов к любому другому базису при линейно входящих коэффициентах.

### 3. ПРОГРАММА ИДЕНТИФИКАЦИИ СЛОЖНОЙ СИСТЕМЫ ДЛЯ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ С ОБЯЗАТЕЛЬНЫМ ВКЛЮЧЕНИЕМ ЗАДАННЫХ АРГУМЕНТОВ-УПРАВЛЕНИЙ

Программа (прогр. 32) предназначена для получения математической модели объекта в виде системы линейных разностных уравнений при помощи специальной процедуры многочленной селекции, гарантирующей наличие в оптимальной модели заданных управляемых воздействий [34].

Язык программирования АЛГОЛ ЭВМ БЭСМ-6.

**Используемые подпрограммы.** GAUSS, SEL.

**Метод.** Предполагается, что общее математическое описание динамики объекта управления имеет вид:

$$\frac{\Delta X_i}{\Delta T} = f_i(X_1, X_2, \dots, X_m, \lambda_1, \dots, \lambda_r, u_1, u_2, \dots, u_r), \quad i = \overline{1, m}, \quad (3.25)$$

где  $X_i$  — одна из системных координат;  $\Delta X_i$  — ее передняя разность;  $X_1, X_2, \dots, X_m$  — множество системных координат (измеренных в текущий момент времени и со всеми заданными запаздываниями);  $\lambda_1, \dots, \lambda_r$  — множество возможных внешних воздействий;  $u_1, u_2, \dots, u_r$  — множество регулирующих воздействий;  $T$  — время (измеряемое дискретно,  $\Delta T = 1$ ).

Программа позволяет выбрать оптимальную структуру модели (с сохранением аргументов-управлений) и находить оптимальную структуру. При выборе структуры используется перебор аргументов группами (или попарно), коэффициенты определяются методом наименьших квадратов на обучающей выборке исходных данных, и полученный вариант описания оценивается по заданному критерию селекции на проверочной выборке данных.

В данной программе заданные  $r$  аргументов-управлений включаются в модель с первого ряда селекции до последнего. При этом в каждом ряду селекции в режиме текущей независимой селекции отбираются по одному наилучшему описанию, содержащему заданный аргумент-управление. Описание, не прошедшие отбор, и те, которые не содержат аргументов-управлений, поступают на общую процедуру селекции, которая отбирает из них  $F$  лучших. Всего на каждом ряду отбирается  $F+r$  описаний.

На последнем ряду селекции  $r$  описаний, содержащих  $r$  заданных аргументов-управлений, объединяются с одним наилучшим описанием из числа  $F$ . Полученное обобщенное описание (состоящее из  $r+1$  описания последнего ряда селекции) обязательно содержит все  $r$  аргументов-управлений, поступающих на общую процедуру селекции, которая ищет в классе линейных полиномов и, соответственно, описание каждого ряда селекции имеет вид

$$y_k = a_0 + a_1 X_1 + a_2 X_2; \quad i = \overline{1, p}; \quad i \neq r; \quad k = 1, \dots, (\overline{F+r}), \quad (3.26)$$

где  $p = m + l + r$  на первом ряду селекции;  $p = F + r$  — на всех последующих рядах;  $m, l$  — число системных координат и возмущающих воздействий. При обучении коэффициентов  $a_0, a_1, a_2$  левые части формулы (3.26) приравниваются передним разностям соответствующей перененной ( $y_k$ ).

Селекция производится по комбинированному критерию — несменчивость plus среднеквадратичная ошибка (СКО). Точки таблицы исходных данных ранжируются в ряд по убывающим значениям дисперсии. Составляется новая последовательность данных, в которой следуют четные точки, взятые послеследовательно из ряда, ранжированного по дисперсии (в новой нумерации от 1 до  $R$ ), а далее — нечетные точки того же ряда (нумеруются от  $R+1$  до  $R1$ ). Формула для оценки по комбинированному критерию имеет следующий вид:

$$\varepsilon_k^2 = \frac{\gamma \sum_{i=1}^{R_1} (y_i^* - y_i^{**})^2 + (1-\gamma) \left[ \sum_{i=R+1}^R (y_i^{**} - y_i)^2 + \sum_{i=R+1}^{R_1} (y_i^* - y_i)^2 \right]}{\sum_{i=1}^{R_1} y_i^2}, \quad (3.27)$$

где  $y_i$  — точки последовательности исходных данных;  $y_i^*$ ,  $y_i^{**}$  — оценки, полученные по формулам, идентифицированным на четных и нечетных точках соответственно;  $\gamma$  — весовой коэффициент (обычно 0,5, но может подбираться в процессе счёта задачи);  $R = R1 = N/2$ ;  $N1$  — число точек наблюдения.

В формуле (3.27) первое слагаемое (при коэффициенте  $1 - \gamma$ ) — показатель смещения, второе (при коэффициенте  $1 - \gamma$ ) — составной показатель СКО, вычисляемый для оценок  $y_i^*$  и  $y_i^{**}$  на группах точек, не участвующих в обучении (проверочных последовательностях). По минимуму указанного выше критерия в каждом ряду селекции выбирается  $R + r$  уравнений, коэффициенты которых уточняются методом наименьших квадратов по всем заданным точкам (этап адаптации коэффициентов).

**Описание параметров.** Входные:  
 МО — количество лучших описаний, передаваемых на последующий ряд селекции;

М — общее количество переменных;  
 Н1 — число точек интерполяции;  
 ТАУ — максимальное число тактов запаздывания;

Л5 — общее количество возмущающих внешних воздействий (включая время  $T$ );  
 Л — количество элементов опорного полинома (при  $L = 3$  полинома вида (3.26));

Л1 — максимальное количество рядов селекции;

КПО — количество описаний из числа МО, для которых выводится на печать показатель  $\varepsilon_k^2$  и значение СКО;

УТ — клоч, указывающий на наличие переменной Т: УТ = 1 — есть Т, УТ = 0 — нет;

УВ — количество первенчих-управлений;

У — количество обзагетных аргументов управлений;

Б — количество вариантов описаний для каждого уравнения при получении обобщенного описания ( $B = Y + 1$ );  
 С2 — коэффициент веса ( $\gamma$  в формule (3.27));

Х1 [1 : N1, 1 : M] — массив исходных данных. Порядок следования переменных: прогностируемые переменные X, измеряемые возмущения (k), управления (u) и время (T),

УЛ [1 : У] — массив номеров обозагетных аргументов управлений (в соответствии с их размещением). Номера упреждающих на один такт значения следуют после номера ( $M = Y$ ) \* ( $1 + TAU$ ) + УT.

**Вычисление:**  
 Х2[1:M\*(1+TAU), 1 : (N1-TAU)] — массив расширенной матрицы исходных данных, в котором западающие аргументы (кроме T) вводятся как самостоятельные (в порядке  $W_0$ ,

$W_{-1}, \dots, W_{-TAU}$ );

LC [1 : M — L5] — массив с указанием оптимального значения числа рядов селекции для прогнозируемых переменных;  
 D51 [1 : MO] — массив лучших показателей  $\varepsilon_k^2$ ;  
 D5 [1 : MO] — массив лучших показателей СКО;  
 ПЛ [1 : Z1, 1 : Б, 1 : M — L5] — массив или запоминания Z1 коэффициентов развернутых полиномов Б для вариантов систем из ( $M-L5$ ) различий уравнений. В первом столбце стоят свободные члены, далее коэффициенты при переменных в том же порядке, что и в массиве X2. В конце массива следуют коэффициенты при упреждениях на один шаг значения уравнений. Программа 32

```

1. 'ALGOL':GAUSS,SEL.,
2. ,BEGIN' STEP,EI'R10,M:11,R,R1,TAY,L5,L1,I,F,KIO,UT,YB,I,J,
3. I11,M2,N,P2,N3,O,P,S,F,T,W1,V2,W,Z,Z1,L11,B,F3,G1,Y,,REAL'
4. C,24,C1,C2,,READ(KO,M,N1,TAY,L5,L,KIO,UT,YB,C2,Y,,,
5. ,BEGIN' INTEGER' ARRAY' LC/1:I-L5/,,,READ(LC) '',
6. 'PRINT',,'//,ЭНД=,,2015)',LC,L1:=0,,FOR I:=1:STEP'1
7. ,UNTIL'M-L5'DO',IP'11,IT'1C/(1/)'THEN'L1:=LC/I/,,Z:=M-YB-YT,,,
8. V2:=M/N+1,,M1:=(M-YT)*V2+YT,,I1:=1-TAY,,Z1:=M-YB+1,,,
9. L11:=2*(I1-1),M2:=M1,,IP'10,GT'M1'THEN'K2:=M0,,,
10. R1:=FHTBKR((I1-V2)/2),,R:=-I1-V2-R1,,PRINT,,,'//,3И MO M,
11. 3ИH N1 TAY L5 L1 KIO UT YB //,1014)',MO,M,N1,TAY,
12. L5,L4,KIO,UT,YB,,,'PRINT',,'//,27H R R1 Z V2 M1 N Z1L1 M2
13. ,/913)',,R,R4,u,V2,M1,N,Z1,L1,M2,,,
14. ,IP'11,ALGAY,X1(1:I1,1:M'),YK(1:Y),XR,XK(1:M2+YB,1:I-L5),,
15. SR(1:M/),X2(1:M*V2,1:W/),Y1(1:M-L5,1:I-1/),AD(1:N-1/),
16. A1(1:L,1:1+1/),A2(1:N-1,1:L+1/),D51(1:MO/),J6(1:L+1/),
17. D5(1:W/),P1(1:L/),D0(1:L+1:L/),E1(1:Z1,1:I-L5/),
18. J,L,D1(1:MO,1:L/),,INTEGER',IP'11,UT,YK(1:Y),,
19. J0(1:MO,1:2/),IP(1:L+1:M,1:2/),II(1:MO+Y,1:2/),,
20. REAK(YL),,PRINT,,,'//,2ИМСИВ ИСЛЯЩИХ X1=,,10Е12.4)',X1,,,
21. 'PRINT',,'//,2ИМСИВ ИСЛЯЩИХ X1=,,10Е12.4)',X1,,,
22. ,FOR I:=1:STEP'1UNTIL'YDO',BEGIN'UMK/1/):=107.,MK/M0+1,1/)
```

```

23. ::=TH(M0+T,2/):=1'END'., 'BEGIN' 'FOR' 'I:=1'STEP'1'UNTIL'M' DO'
24. 'BETH' 'SRC(I/):=0., 'FOR' 'J:=1'STEP'1'UNTIL'N1' DO' 'SR(I/):=
25. 'SR(I/)+K(J,I,J)AL1'END'., 'PRINT' , 'C//,8UCPTEUE ,
26. 'BHEHNEHLM MACCB SR.,//,8E15,8)' , SR.. 'BEGIN' 'FOR' 'I:=1
27. 'STEP'1'UNTIL'M' DO' 'FOR' 'T:=0' STEP'1'UNTIL'TAY' DO'
28. 'BETH' 'FOR' 'J:=(TAY+1)-T' STEP'1'UNTIL'(N1-T)' DO'
29. 'ZC((I*(TAY+1)-TAY+T),(J-TAY+T)):=X1((J,I/)-SR(I/))/
30. 'SR((J,I/))END'., 'FOR' 'J:=1'STEP'1'UNTIL'M-1' DO' BEGIN' 'FOR' 'I:=1
31. 'STEP'1'UNTIL'N-1' DO' Y1((J,I/)):=ZC((J-1)*(TAY+1)+1,I+1)/
32. -ZC((J-1)*(TAY+1)+1,I/))END'., 'FOR' 'I:=1'STEP'1'UNTIL'N-1
33. 'DO' , 'BEGT' 'AD(I/):=0., 'FOR' 'J:=1'STEP'1'UNTIL'M1' DO' W((J,I/)):=
34. 'AD((J,I/)+X2((J,I/)*2)END'., 'PRINT' , 'C//,18HUNOER TUKH , MCHB,
35. 'SHOHL H-C=) .,, 'FOR' 'I:=1' STEP'1'UNTIL'N-1' DO' 'BEGIN' H((J,I/)):=
36. 'I..C:=AD(I/),.. 'FOR' 'J:=1'STEP'1'UNTIL'N-1' DO' 'IF' C'LT' AL(J/)/
37. 'THEN' 'BETH' 'C:=AD(J/),.. H((J,I/)):=J.,.. END'., 'ZC((1,I/)):=0.,
38. 'AD((J,I/)):=0'END'., 'PRINT' , '(6,16,E14,5))' , 'FOR' 'I:=1'STEP'1
39. '1'UNTIL'N-1' DO' H((J,I/),A2((1,I/))),.. 'FOR' 'M:=1'STEP'1'UNTIL'
40. 'N-L5' DO' 'BEGIN' 'IF' 'M=2' THEN' Y:=0., 'PRINT' , 'C//,6HUNOEP ,
41. '11M#05 YAPBL=,12)',.. 'M..,L1:=LG((M/),..,DA:=0., 'FOR' 'I:=1'STEP'
42. '1'UNTIL'N' DO' DM:=D4+y1((M1,1C(J/)))*2., 'FOR' 'I:=M-R1' STEP'1
43. 'UNTIL'N-1' DO' DM:=D4+y1((M1,1C(J/)))*2., 'PRINT' , 'C//,3M#A=,
44. E14,5)',.. DA., 'FOR' 'J:=1'STEP'1'UNTIL'N-1' DO' 12((J,L+1)/):=
45. Y1((J,M1,1C(J/)).., 'FOR' 'I:=1'STEP'1'UNTIL'M1' DO' 'FOR' 'J:=1'STEP'
46. '1'UNTIL'N-1' DO' XP((J,I/)):=ZC((1,M1,I,J)).., 'FOR' 'T:=1'STEP'1
47. 'UNTIL'YB' DO' 'FOR' 'J:=1'STEP'1'UNTIL'N-1' DO' XPK(M1+I,J):=
48. 'X2((M-YB-YT+I-1)*V2+1,H((J/Y)+1/),.. P:=M1+YB., 'FOR' 'T:=1'STEP'
49. '1'UNTIL'N-1' DO' 'XP((J,I/)):=ZC((1,M1,I,J)).., 'IF' T GT 1 THEN' P:=M0.,
50. 'FOR' 'I:=1'STEP'1'UNTIL'P-1' DO' 'FOR' 'J:=I+1'STEP'1'UNTIL'P' DO'
51. 'SEL(I,J,X,P,D51,M1,M2,N,H2,M3,1H,R,R1,I,M0,Y,YL,YM,C2,D4,..)
52. 'IF' T EQ 1' THEN' 'FOR' 'I:=1'STEP'1'UNTIL'P' DO' YL((I/)):=M0-Y+I.,
53. 'FOR' 'I:=y+1'STEP'1'UNTIL'P-1' DO' 'FOR' 'J:=I+1'STEP'1'UNTIL'P' DO'
54. 'H((I-Y,2/):=TH((I,2/),.. D51((I-Y/)):=D51((I/))END'.,
55. 'FOR' 'I:=1'STEP'1'UNTIL'Y' DO' 'BEGIN' D51((M0-Y+I/)):=YL((I/)),
56. 'TH((M0-Y+I,1/):=TH((M0+I,1/),.. D51((M0-Y+I,2/)):=TH((M0+I,2/),
57. 'END'., 'FOR' 'I:=1'STEP'1'UNTIL'Y' DO' 'BEGIN' YM((I/)):=107.,

58. TH((M0+I,1/)):=TH((M0+I,2/)):=1'END'., 'PRINT' , 'C//,6HPTA N=,12,
59. 'C//,12HERROR GAUSS=,13,5X,12HUREPESOP HAP=,14)' , T,M2,N3.,
60. 'FOR' 'J:=1'STEP'1'UNTIL'M0'DO' 'BEGT' 'FOR' 'J:=1'STEP'1'UNTIL'
61. 'N-1' DO' 'BEGT' 'M2((J,1/):=1.. A2((J,2/)):=XP((H((J,1,J/),J/)),.
62. 'A2((J,3/)):=XP((H((J,2,J/)),J/))END'., 'FOR' 'J:=1'STEP'1'UNTIL'L
63. 'DO' , 'FOR' 'T:=1'STEP'1'UNTIL'L+1' DO' H((J,F/)):=0., 'FOR' 'J:=1
64. 'STEP'1'UNTIL'L' DO' 'BEGT' 'FOR' 'F:=1'STEP'1'UNTIL'L+1' DO'
65. 'FOR' 'O:=1'STEP'1'UNTIL'N-1' DO' A1((J,F/)):=A1((J,F/))+H2((0,I/))
66. 'A2((0,F/))EID'., 'GAUSS(M1,L,Y1,ERROR)., 'ERROR':D5((1/)):=0.,
67. 'FOR' 'J:=1'STEP'1'UNTIL'N-1' DO' 'BEGT' 'C:=0., 'FOR' 'F:=1'STEP'1
68. 'UNTIL'L'DO' C:=G+H((F/))*N2((J,F/),.. XII((1,J/)):=C., 'D5((1/)):=
69. 'D5((1/))+((2((J,L+1))-C))*2/M4'END'., 'FOR' 'J:=1'STEP'1'UNTIL'L
70. 'DO' , 'DGT((G-1)*M0+1,J/)):=P1((J/)),.. 'FOR' 'J:=1'STEP'1'UNTIL'L
71. 'DO' , 'DGT((T-1)*M0+1,J/)):=TH((J,J/))END'., 'FOR' 'I:=1'STEP'1
72. 'UNTIL'M0'DO' 'FOR' 'J:=1'STEP'1'UNTIL'N-1' DO' KP((T,J/)):=XI((1,
73. 'J/)),.. 'FOR' 'J:=M0-K0+1'STEP'2'UNTIL'M0'DO' 'PRINT' , '(2(2X,14,
74. '2E20,5,110,15))' , 'FOR' 'K:=0,1' DO' ((+K,D51((J+K/)),D5((J+K/),
75. 'H((J+K,1/),H((J+K,2/)))END'., 'FOR' 'K:=1'STEP'1'UNTIL'K'DO'
76. 'BEGT' 'FOR' 'I:=1'STEP'1'UNTIL'Z1'DO' IM((I,T,K,MW)):=0., 'FOR' 'I:=
77. '1'STEP'1'UNTIL'2'DO' 'FOR' 'J:=1'STEP'1'UNTIL'L11M'DO' IM((J,1)
78. ':=0., 'FOR' 'I:=1'STEP'1'UNTIL'L'DO' 'FOR' 'J:=1'STEP'1'UNTIL'
79. 'M0'DO' DK((J,I/)):=DH((J,I/)):=0., HM((1,2/)):=M0-K+1., 'FOR'
80. 'I:=1'STEP'1'UNTIL'2'DO' 'FOR' 'J:=1'STEP'1'UNTIL'L11M'DO' IM((J,1)
81. 'FOR' 'O:=L1-1'STEP'-1'UNTIL'1'DO' 'BEGT' 'R:=0., 'FOR' 'I:=1'STEP',
82. '1'UNTIL'L11M'DO' 'PR' IM((1,2/)):=IM((1,K,MW)):=IM((1,K,MW)-DK1((N3,1)),
83. 'N3:=IM((1,2/)),.. IM((1,K,MW)):=IM((1,K,MW)-DK1((N3,1)),
84. 'P:=IM((N2,1/)):=D0((M0*0+H3,1/)),.. N2:=P+2+1., S:=IM((N2,1/)):=
85. 'IM((M0*0+H3,2/)),.. 'FOR' 'J:=1'STEP'1'UNTIL'L'DO' 'BEGT'
86. 'DK((P,J/)):=DK((P,J/))+DK((O-1)*M0+P,J/)*DK1((N3,2/)),
87. 'DK((S,J/)):=DK((S,J/))+DK((O-1)*M0+S,J/)*DK1((N3,3/))END',
88. 'IND'., 'FOR' 'J:=1'STEP'1'UNTIL'L11M'DO' 'BEGT' 'IM((J,2/)):=
89. 'IM((J,1/)),.. IM((J,1/)):=0'END'., 'FOR' 'J:=1'STEP'1'UNTIL'L'DO'
90. 'FOR' 'P:=1'STEP'1'UNTIL'M0'DO' 'BEGIN' DK1((P,J/)):=DK((P,J/)),
91. 'IM((P,J/)):=0'END'., 'FOR' 'I:=1'STEP'1'UNTIL'L11M'DO'
92. 'IF' YM((I,2/))NE 0' THEN' 'BEGIN' N3:=IM((1,2/)),.. IM((1,K,MW)):=
93. 'IM((1,K,MW)):=IM((1,M3,1/)),.. IM((M0*N3,1/)+1,K,MW)):=

```

```

94.  ИЛ(/ДО(/N3,1/)+1,K,ИЛ/)+ДИ1(/N3,2/):=ИЛ(/ДО(/N3,2/)+1,K,ИМ/)

95.  :=ИЛ(/ДО(/N3,2/)+1,K,ИМ/)+ДИ1(/N3,3/)*'END'., 'FOR' I:=1'STEP'
96.  1'UNTIL'Z1-1'DO'ИЛ(/1,K,ИМ/):=ИЛ(/1,K,ИМ/)-ИЛ(/I+1,K,ИМ/),'
97.  'FOR' I:=1'STEP'1'UNTIL'I'M-УТ'DO' 'FOR' J:=1'STEP'1'UNTIL'V2'DO',
98.  ИЛ(/(I-1)*V2+J+1,K,ИМ/):=ИЛ(/(I-1)*V2+J+1,K,ИМ/)/SR(/T/),
99.  'IF'УТ'EQ'1'THEN'ИЛ(/Z1-YB,K,ИМ/):=ИЛ(/Z1-YB,K,ИМ/)/SR(/M/),
100. 'FOR' I:=1'STEP'1'UNTIL'УВ'DO'ИЛ(/Z1-YB+I,K,ИМ/):=
102. ИЛ(/Z1-YB+I,K,ИМ/)/SR(/M-(УВ+УТ)+I/), 'FOR' I:=1'STEP'1'UNTIL'
103. Z1'DO'ИЛ(/1,K,ИМ/):=ИЛ(/1,K,ИМ/)*SR(/M/)'END'., 'FOR' J:=1
104. 'STEP'1'UNTIL'Z1'DO' 'BEGIN'ИЛ(/J,1,ИМ/):=ИЛ(/J,1,ИМ/)'Б..,
105. 'FOR' K:=2'STEP'1'UNTIL'Б'DO'ИЛ(/J,1,ИМ/):=ИЛ(/J,1,ИМ/)+'
106. ИЛ(/J,K,ИМ/)'Б..END', 'PRINT'', '(10E12.5)', 'FOR' J:=1'STEP'1
107. 'UNTIL'Z1'DO'ИЛ(/J,1,ИМ/)'END'END', 'END', 'END', 'END', 'EOP'

ПОДПРОГРАММА SEL

```

Селекция МО лучших описаний по критерию  $\varepsilon_k^2$ . Отдельно селектируется по одному лучшему описанию, содержащему заданный аргумент-управление (прогр. 33)

*Обращение* SEL (I, J, XP, D51, A1, A2, N, N2, N3, IH, R, R1, L, MO, Y, YL, YM, C2, D4).

*Описание параметров:*

- I, J — номера пары исходных или промежуточных аргументов;
- XP [1 : M2+YB, 1 : N-1] — массив значений исходных или промежуточных переменных;
- D51, L, MO, Y, YL, C2 — описанные выше;
- A1, A2, N, N2, N3 — массивы и идентификаторы, используемые при промежуточных вычислениях в основной программе.

*Метод* Дважды вычисляются по МНК коэффициенты каждого частного описания, на двух частях выборки исходных данных. Полученные модели оцениваются по критерию  $\varepsilon_k^2$ , по которому отбираются лучшие описания!

### ПРОГРАММА 33

```

1.  'ALGOL':GAUSS.,
2.  'PROCEDURE'SEL(I,J,XP,D51,A1,A2,N,N2,N3,IH,R,R1,L,M0,Y,YL,YM,
3.  C2,D4),,'INTEGER' I,J,N,N2,N3,R,R1,L,M0,Y,,ARRAY'XP,D51,A1,A2,
4.  YM,'INTEGER'ARRAY'IH,YL,'REAL' C2,D4,, 'BEGIN' INTEGER'K,F,
6.  H,S,, 'REAL'W,W1,W2,, 'ARRAY'P,P2(/1:L/),,'INTEGER' I1,J1,
7.  'FOR' H:=1'STEP'1'UNTIL'H-1'DO' 'BEGIN' A2(/H,1/):=1,,A2(/H,2/),
8.  ::'XP'(/1,H/),,A2(/H,3/):=XP(/J,H/)'END'., 'FOR' K:=1'STEP'1
9.  'UNTIL'L'DO' 'FOR' F:=1'STEP'1'UNTIL'L+1'DO'A1(/K,F/):=0., 'FOR'
10. K:=1'STEP'1'UNTIL'L'DO' 'BEGIN' FOR' F:=1'STEP'1'UNTIL'L+1'DO'

11. 'FOR' H:=1'STEP'1'UNTIL'R'DO'A1(/K,F/):=A1(/K,F/)+A2(/H,K/)*
12. A2(/H,F/)'END'., 'GAUSS'(A1,L,P1,ERROR),,'FOR' K:=1'STEP'1
13. 'UNTIL'L'DO' 'FOR' F:=1'STEP'1'UNTIL'L+1'DO'A1(/K,F/):=0.,
14. 'FOR' K:=1'STEP'1'UNTIL'L'DO' 'BEGIN' 'FOR' F:=1'STEP'1'UNTIL'L+1
15. 'DO' 'FOR' H:=N-R1'STEP'1'UNTIL'N-1'DO'A1(/K,F/):=A1(/K,F/)+
16. A2(/H,K/)*A2(/H,F/)'END'., 'GAUSS'(A1,L,P2,ERROR),, W:=0.,
17. 'FOR' K:=1'STEP'1'UNTIL'R'DO' 'BEGIN' W1:=0.,W2:=0., 'FOR' F:=1
18. 'STEP'1'UNTIL'L'DO' 'BEGIN' W1:=W1+P1(/F/)*A2(/K,F/),,W2:=W2+
19. P2(/F/)*A2(/K,F/)'END'., W:=W+(W1-W2)**2'END'., 'FOR' K:=N-R1
20. 'STEP'1'UNTIL'N-1'DO' 'BEGIN' W1:=W1+P1(/F/) A2(/K,F/),,W2:=W2+P2(/F/)*
21. 'UNTIL'L'DO' 'BEGIN' W1:=W1+P1(/F/) A2(/K,L+1/)*2'END'., W:=W+C2., 'FOR' K:=1
22. A2(/K,F/)'END'., W:=W+(W1-W2)**2'END'., W:=W+C2., 'FOR' K:=1
23. 'STEP'1'UNTIL'R'DO' 'BEGIN' W1:=0., 'FOR' F:=1'STEP'1'UNTIL'L'DO'
24. W1:=W1+P2(/F/)*A2(/K,F/),,W:=W+((W1-A2(/K,L+1/))*2)*(1-C2)
25. 'END'., 'FOR' K:=N-R1 'STEP'1'UNTIL'N-1'DO' 'BEGIN' W1:=0.,
26. 'FOR' F:=1'STEP'1'UNTIL'L'DO' W1:=W1+P1(/F/)*A2(/K,F/),
27. W:=W+((W1-A2(/K,L+1/))*2)*(1-C2)'END'., W:=W/D4.,I1:=I.,
28. J1:=J., 'FOR' K:=1'STEP'1'UNTIL'Y'DO' 'BEGIN' IF'IEQ'YIK/K/
29. ,OR'J'EO'YIK/K/)'THEN' 'BEGIN' IF'W'LT'YMK/K/)'THEN' 'BEGIN',
30. W1:=W1-ИЛ(/W1-A2(/K,L+1/))**2*(1-C2)'END'., W:=W/D4.,I1:=I.,
31. ИЛ(/M0*K,4/):=I.,I:=I.,H:=ИЛ(/M0*K,2/),,ИЛ(/M0*K,1/):=J..
32. J:=H., 'GOTO'M55'END'., 'GOTOM55'END'END'., M55:N3:=N3+1.,
33. 'IF'N3'LEMO'THEN' 'BEGIN' D51(/N3/):=W.. ИЛ(/N3,1/):=I.,
34. ИЛ(/N3,2/):=J., 'IF'N2=M0'THEN' 'BEGIN' 'FOR' K:=M0 'STEP'-1
35. 'UNTIL'2'DO' 'BEGIN' W1:=D51(/K/),,H:=K.. 'FOR' F:=K-1'STEP'-1
36. 'UNTIL'1'DO' 'IF'D51(/F/)'LT'W1'THEN' 'BEGIN' W1:=D51(/F/),,H:=F
37. 'END'., 'IF' H'NEq' R'THEN' 'BEGIN' D51(/H/):=D51(/K/),
38. D51(/K/):=W1.,S:=ИЛ(/H,1/),,ИЛ(/J,1/):=ИЛ(/R,1/),
39. ИЛ(/R,1/):=S.. S:=ИЛ(/H,2/),,ИЛ(/H,2/):=ИЛ(/K,2/),,ИЛ(/K,2/),
40. ::S'END'END'END'END'ELSE'BEGIN' IF'W'LT'D51(/M0/),THEN' 'BEGIN' S:=M0.,
41. 'GOTO'N5'ELSE'BEGIN' IF'W'LT'D51(/M0/),THEN' 'BEGIN' S:=M0.,
42. 'GOTO'M3'END'.,S:=0.,W4:S:=S+1., 'IF'W'LT'D51(/S+1/)'THEN'
43. 'GOTO'M4..M3:=ИЛ(/1,1/):=I.,ИЛ(/1,2/):=J.,D51(/1/):=W.,
44. 'FOR' K:=S 'STEP'-1'UNTIL'2'DO' 'BEGIN' W1:=D51(/1/),,D51(/1/):=
45. D51(/K/),,D51(/K/):=W1.,H:=ИЛ(/1,1/),,ИЛ(/1,1/):=ИЛ(/K,1/),,
```

46.  $\text{IN}(/R,1/)=N, \text{IN}=\text{IN}(1,2)/, \text{IN}(1,2/)=\text{IN}(/R,2/), \text{IN}(/R,2/)=$   
 47.  $\text{N}=\text{END}, \text{END}, \text{END}, \text{GOTO} M5, \text{ERROR}: N2:=N2+1, M5: I:=I1, J:=J1$   
 48.  $\text{END}, \text{EOF}$

## С ПРИМЕНЕНИЕМ КОМБИНАТОРНОГО ПЕРЕБОРА ЧАСТНЫХ ОПИСАНИЙ

Предназначается (прогр. 34) для математического описания сложных объектов и реализует многорядный селекционный отбор с использованием комбинаторного перевора частных описаний [23].

Язык программирования FORTRAN, ЭВМ EC-1022.

Используемые подпрограммы: REGR, ZAR, NORM, CORA, REGA, MISR, MISN, KRIT, MAII, RAZR, DICH.

Метод. Полное частное описание имеет вид

$$z = a_1 + a_2 x_1 + a_3 x_2 + a_4 x_1 x_2. \quad (3.28)$$

Селекция частных описаний на каждом ряду может происходить по значениям пяти видов критерии регулярности, двум критериям несмешенности и двум комбинированным критериям.

Программа позволяет использовать различные модификации основного алгоритма в зависимости от выбранного пользователем способа конкуренции частных описаний, правила останова, вида критерия и способа заполнения входного массива для каждого ряда селекции.

**Модификации программы.** Выполнение программы зависит от значений следующих параметров (табл. 5):

Таблица 5

Пара- метр	Возможные значения	Стандартный режим		Стандартный режим программы
		Пара- метр	Возможные значения	
N	> 1, ≤ NN	N = NN	≤ NDOB, ≥ 3	V = 10
M	> 1, ≤ MM	M = MM	NRED ≥ 1	NRED = 5
N1	≥ M4, < N	N1 = 0,7 * N	LK = 1	LK = 1
N2	≥ M4, < N	N2 = N/2	NPROC = 1	NPROC = 1
LM	1, 2, 3, 4, 5, 6, 7, 8, 9	LM = 1	KLU = 0	KLU = 0

LM — задает вид используемого критерия селекции. LM = 1; 2; 3; 4; 5 соотвествует различным видам критерия регулярности (см. подпрограмму MISR); LM = 6; 7 соответствует двум видам критерия несмешенности (см. подпрограмму MISN); LM = 8; 9 соответствует двум комбинированным критериям (см. подпрограмму KRIT);

LK — определяет способ заполнения входного массива для следующего ряда селекции. Если LK = 1; 2, 3, 4, массив ZK заполняется значениями NPROC — максимальное число рядов селекции;

NRED — число частных описаний, пропускаемых на каждом ряду;

KLУ — параметр, задающий способ заполнения входного массива для следующего ряда селекции, определяющий способ конкурирования частных описаний;

NPROC — параметр, позволяющий исключать (KLU = 1) или оставлять (KLU = 0) множественный регрессионный анализ в программе для сравнения с селекционным алгоритмом;

ZN (NN, MM) — массив исходных данных для каждого ряда селекции;

X (NN, MM4) — рабочий массив, используемый при комбинаторном переборе;

XС (MM4) — вектор средних значений переменных X (NN, MM4);

SX (MM4) — вектор средневариационных отклонений переменных массива X (NN, MM4);

X1 (NN, MM4) — нормированные значения массива X;

Z (MM4, MM4) — матрица коэффициентов парной корреляции переменных массива X1;

A (MM4), AA (MM4), AAB (MM4) — векторы коэффициентов частных описаний, вычисленных на точках, лежащих соответственно, в пределах [1, N1], [1, N2], [(N2+1), N], [1, N];

REG (LL) — вектор значений критерии селекции;

DOSH (NDOB) — вектор значений заданного критерия селекции для лучших частных описаний ряда;

DPS (NDOB, M4) — массив коэффициентов V лучших частных описаний линейного ряда;

IV (NDOB, 4) — массив основных характеристик лучших частных описаний данного ряда. Расшифровка по столб-

ции отбираются как из частных описаний текущего ряда, так и из лучших описаний предыдущих рядов. Селекция будет продолжаться до тех пор, пока среди лучших описаний не окажется ни одного стесненного на данном ряду. При этом число рядов менять не нужно.

### Описание параметров:

MM, NN — максимальное число переменных и точек соответственно;

LL — максимальное число критерииев;

NDOB — максимальное число лучших описаний, пропускаемых в каждом ряду;

MM4 — максимальное число коэффициентов в полном частном описании;

M4 — число коэффициентов в полном частном описании: M4 = M4—1;

NK — число частных описаний функций двух аргументов при комбинаторном переборе ( $NK = \sum_{i=1}^{M4-1} C_{M4-i}$ ),

M, N — число переменных и точек для данной задачи; N1 — число точек в обучающей последовательности, используемой в критерии регулярности ( $N > N1 \geq M4$ ). Для проверочной последовательности берутся точки от  $N1 + 1$  до  $N$ ;

N2 — число точек в обучающей и проверочной последовательностях при использовании критерия несмешенности;

LM — номер используемого критерия;

V — число частных описаний, пропускаемых на каждом ряду;

NRED — максимальное число рядов селекции;

KLУ — параметр, задающий способ заполнения входного массива для следующего ряда селекции;

NPROC — параметр, определяющий способ конкурирования частных описаний;

REG (LL) — вектор значений критерии селекции;

DOSH (NDOB) — вектор значений заданного критерия селекции для лучших частных описаний ряда;

DPS (NDOB, M4) — массив коэффициентов V лучших частных описаний линейного ряда;

IV (NDOB, 4) — массив основных характеристики лучших частных описаний данного ряда. Расшифровка по столб-

чам: I, II — номера аргументов данного частного описания; III — номер типа частного описания, IV — номер ряда селекции, на котором данное описание получено;

ID (MM4) — бинарный вектор, определяющий структуру частного описания;

IIP, IQ — номера аргументов частного описания;

W — номер ряда селекции;

iQQ — номер типа частного описания (табл. 6);

**Типы частных описаний**

Частное описание	
1	$y_1 = a_1 + a_2x_1x_2$
2	$y_2 = a_1 + a_2x_2$
3	$y_3 = a_1 + a_2x_2 + a_3x_1x_2$
4	$y_4 = a_1 + a_2x_1$
5	$y_5 = a_1 + a_2x_1 + a_3x_1x_2$
6	$y_6 = a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2$
7	

**Таблица 6**

V лучших частных описаний данного ряда;  
оформление результатов счета данного ряда;  
проверка правильности останова и перехода на следующий ряд селекции.

**Подготовка исходных данных.** Все входные параметры и исходные данные вводятся с помощью оператора NAMELIST (PARLM — блок параметров, DANNI — блок данных). Обязательно задавать значения только тех параметров, которые для конкретной задачи имеют значения, отличные от стандартных (см. табл. 5).

**Выход на печать.** В блоке PARLM выводится массив исходных данных ZN (N, M). В блоке ROKOL для каждого ряда селекции выводится: номер ряда; чисто отобранных лучших частных описаний; максимальное, среднее и минимальное значение критерия. Для каждого из V отобранных частных описаний выводится: 1-й столбец — номер частного описания этого ряда селекции; 2-й и 3-й столбцы — номера аргументов предыдущего ряда, которые входят в частное описание; 4-й столбец — номер типа частного описания; 5-й столбец — номер ряда селекции, на котором было сгенерировано частное описание; 6-й, 7-й, 8-й и 9-й столбцы — коэффициенты частного описания, 10-й столбец — значение заданного критерия.

Если LK = 5, то пересчитываются коэффициенты по линиям всей последовательности при заданной структуре частного описания, и новые значения коэффициентов и критерии выводятся на печать. Далее печатаются вычислительные промежуточные переменные, полученные на основе V лучших частных описаний.

### 1.С МИТОРИДЫЙ СЕЛЕКЦИОННЫЙ АЛГОРИТМ С ПРИМЕНЕНИЕМ КОМПЬЮТЕРНОГО ПЕРЕВОРА ЧАСТНЫХ ОПИСАНИЙ

<pre> 4. INTEGER T,V,W,IU(3),IV(50,4) 5. REAL ZH(100,50),ZH(100,50),DPS(50,4),DOSH(50), 6. X(100,50),X(100,50),Z(50,50),R(50,50),F(50),R(50), 7. A(50),XC(50),SX(50),UM(100),UB(100),UAB(100), 8. REC(10),AAC(50),AB(50),AAB(50),U(100) 9. DATA N#M,IJ,NDOB,M#M,M#I,NK / 10. *100,50,10,50,50,4,3,7,/ 11. COMMON /KEY/ KLU 12. NAMELIST /PARAM/ N,M,N1,N2,IJ,V,NRED,KLU,IK,NPROC 13. NAMELIST /DANNI/ ZN 14. NAMELIST /OCENKI/ RR,REC,SX 15. NAMELIST /ROKOL/ W,V,SY,SSR,SM 16. С ПАРАМЕТРЫ ПРОГРАММЫ 17. DATA IJ,IK,NPROC,V,NRED /1,1,1,10,5/ 18. M#I=M#I-1 19. N#N 20. M#M </pre>
---

```

21. READ(1,PARAM)
22. IF(N1.EQ.0) N1=0.7*N
23. IF(N2.EQ.0) N2=N/2
24. WRITE(3,PARAM)
25.C           ВВОД ДАННЫХ
26. DO 302 I=1,48
27. READ(1,300,ERR=303) NT,NK,NR,(ZN(I,K),K=1,13)
28. WRITE(3,300)      NT,NK,NR,(ZN(I,K),K=1,13)
29. GOTO 302
30. 303 PAUSE
31. 302 CONTINUE
32. DO 305 I=1,48
33. READ(1,301,ERR=306) NT,NK,NR,(ZN(I,K),K=14,23)
34. WRITE(3,301)      NT,NK,NR,(ZN(I,K),K=14,23)
35. GOTO 305
36. 306 PAUSE
37. 305 CONTINUE
38. 300 FORMAT(2I2,11,F7.0,3F8.0,3F5.2,3F5.0,2F5.2,F4.2)
39. 301 FORMAT(2I2,11,5F5.0,4F5.2,F8.0)
40. 201 FORMAT(1X,10G12.6,/)

41. KLU=1
42. MC=M
43. NV=L
44.C           ИНТОРИДНАЯ СЕЛЕКЦИЯ
45.C
46. DO 21 I=1,NDUB
47. 21 DUSH(I)=9999999999. '
48. DO 101 W=1,NRED
49.C           СПОСОБЫ КОНФИГУРИРОВАНИЯ ЧАСТНЫХ ОПИСАНИЙ
50. IF(NP0C.EQ.0) GOTO 23
51. DO 25 I=1,NDUB
52. 25 DUSH(I)=9999999999.
53. 23 CONTINUE
54. NV1=NV-1
55.C           ГЕНЕРИРОВАНИЕ ПАРМ
56. DO 1 IIP=2,NV1

57. IIP=IIP+1
58. DO 1 IC=IP1,NV
59.C           КОММУНИКАТОРНЫЙ ПЕРЕВОР
60. DO 41 ICO=1,NK
61. DO 501 T=1,N
62. X(T,1)=ZN(T,1)
63. 501 CONTINUE
64. IF((IO0.EQ.2).OR.(IO0.EQ.4)).AND.(IIP.NE.2)) GOTO 41
65. IF((IO.GT.3).AND.(IO0.EQ.4)) GOTO 41
66. CALL ZAR(N,NM,MN,MU,ZN,X,LD,IIP,IO,IO0,MW)
67. M=MV
68.          КРИТЕРИИ РЕГУЛЯРНОСТИ
69. IF((LM.GT.5).AND.(LM.LT.8)) GOTO 505
70. CALL NORM(1,N1,N,MW,NW,MV,X,X1,XC,SX)
71. CALL CORA(1,N1,N,MV,NW,MV,X1,Z)
72. CALL REGA(MV,MV,XC,SX,Z,B,F,R,AA,RR,KGA)
73. IF(KGA.EQ.0) GOTO 10
74. CALL MISRN1+1,N,H,MV,HN,MV,L1,X,A,Y,REG)
75. 505 CONTINUE
76. IF(LM.LT.G) GOTO 506
77. CALL NORM(1,N2,H,M,HN,MV,X,X1,XC,SX)
78. CALL CORA(1,N2,H,M,MV,MV,X,X1,Z)
79. CALL REGA(M,MV,XC,SX,Z,B,F,R,AA,RR,KGA)
80. IF(KGA.EQ.0) GOTO 10
81. CALL NORM(1,N2+1,N,N,;M,MV,MV,X,X1,XC,SX)
82. CALL CORA(1,N2+1,N,N,M,MV,X,X1,Z)
83. CALL REGA(M,MV,XC,SX,Z,B,F,R,AB,RR,KGA)
84. IF(KGA.EQ.0) GOTO 10
85. CALL NORM(1,N,N,M,MV,MV,X,X1,XC,SX)
86. CALL CORA(1,N,N,M,MV,MV,X,X1,Z)
87. CALL REGA(M,MV,XC,SX,Z,B,F,R,AB,RR,KGA)
88. IF(KGA.EQ.0) GOTO 10
89. CALL MISRN1,MV,HN,MV,L1,X,A,Y,AB,AB,YA,YB,YAB,REG)
90. 506 CONTINUE
91. IF(LM.GT.7) CALL KRIT(L1,REG)

```

```

92. CALL MATR(DOSH,V,SMX,TMX)
93. IF(PREG(MA).GE.SMX) GOTO 4
94. IV(TMX,1)=IP
95. IW(TMX,2)=IO
96. IV(TMX,3)=TOO
97. IV(TMX,4)=I
98. DOZK(TMX)=REG(MA)
99. MV1=IV
100. CALL RAZR(MV1,MM,MV,MV1,1D,A)
101. D0 26 I=1,MV
102. 26 DPSK(TM,IK)=A(IK)
103. D0 9 T=1,N
104. TF(LK,LK,1) ZK(T,TMX)=Y(T)
105. TF(LK,LK,2) ZK(T,TMX)=Y(A(T))
106. TF(LK,LK,3) ZK(T,TMX)=YB(T)
071. TF(LK,LK,4) ZK(T,TMX)=YAK(T)
108. 4 CONTINUE
109. GOTO 11
110. 10 CONTINUE
111. 5 FORMAT(//,'НЕТ РЕШЕНИЯ',//)
112. 11 CONTINUE
113. 41 CONTINUE
114. M=MG
115. 1 CONTINUE
116. MV=V
117. CALL EUSTR(DOSH,V,SMV,SMX,SSR,TMV,TMX)
118.C          (ФОРМИРОВАНИЕ РЕЗУЛЬТАТОВ
119. WRITE(3,POKOL)
120. DO 29 I=1,V
121. WRITE(3,221),(IV(I,IK),IK=1,4),(DPSK(I,K),K=1,MV),
122. *DOSH(I)
123. 22 FORMAT(5I4,6G15.6)
124. 25 CONTINUE
129. C      ИЗРЕЧЕНИЕ КОДОВ УРАВНЕНИЯ С ЛУЧШИМИ СТРУКТУРАМИ
126. 1F(LK,NE,5) GOTO 28

```

127. WRITE(3,32)

132. 32 FORMAT(//,T20,'ИЗРЕЧЕНИЕ КОЭФФИЦИЕНТОВ',//)

129. DO 6 I=1,V

130. CALL ZARK(N,N,MM,MV,MV1,ZN,X,1D,IV(1,1),IV(1,2),

131. \*IV(1,3),MV)

132. CALL NORM(1,N,N,MV,UN,MM,X,X1,XG,SX)

133. CALL CORA(1,N,N,MV,MM,XC,SX,Z,B,F,R,A,RR,KGA)

134. IF(KGA.EQ.0) GOTO 6

135. CALL REG(A,EQ,0) GOTO 6

136. CALL MISR(1,N,N,MV,UN,MM,LL,X,A,Y,REC)

137. D0 8 T=1,N

138. ZK(T,1)=Y(T)

139. 8 CONTINUE

140. MV1=MV

141. CALL RAZR(MV1,MM,MV,MV1,1D,A)

142. WRITE(3,22) I,(IV(I,IK),IK=1,4),(A(IK),K=1,MV),REC(MV)

143. 6 CONTINUE

144. 28 CONTINUE

145.C НОВАЯ ТАБЛИЦА

146. D0 555 T=1,N

147. D0 555 I=1,NJOB

148. ZK(T,I)=ZK(T,I)

149. 555 CONTINUE

150. WRITE(3,201)((ZK(T,I),T=1,N),I=1,V)

115.C ПРАВОДОК ОСТАНОВА

152. D0 30 I=1,V

153. IF(IV(I,4).EQ.W) GOTO 103

154. 30 CONTINUE

155. GOTO 102

156. 103 CONTINUE

157. PAUSE

158. 101 CONTINUE

159. 102 CONTINUE

160. STOP

161. END

## ПОДПРОГРАММА REGR

Расчет коэффициентов линейной регрессии и критерии оценок уравнения регрессии (прогр. 35).

*Обращение.* CALL REGR(N1, N2, N, M, NN, MM, LL, X, X1, XC, SX, Z, B, F, R, A, RR, REG, KGA).

*Описание параметров:*

УА (NN) — вектор вычисленных значений зависимой переменной 10 УР21в

печати регрессии;

В (NN, MM), R (MM) — рабочие массивы;

KGA — признак наличия решений систем нормальных уравнений

(равен 0, если решение системы существует, и 1 — в противном

случае). Значения остальных параметров такие же, как и в

основной программе.

*Требуемое подпрограммы и функции.* NORM, CORA, REGA, MISR, SQRT

\*F(MM), R(MM), A(MM), U(NN), REG(LL)

NORM, которая вычисляет средние значения переменных, их дисперсии, а также

нормированные по линии регрессии значения в отклонениях от средних

Ограничение к подпрограмме CORA, которая вычисляет коэффициенты парной

корреляции для  $i, j = 1, 2, \dots$  по формуле

$$Z_{ij} = \begin{cases} \frac{\sum\limits_{k=N1}^{N2} X_k * X_{ki}}{N2 - N1 + 1}, & j \neq i; \\ 1, & j = i. \end{cases} \quad (3.29)$$

Обращение к подпрограмме REGA, которая вычисляет вектор коэффициентов регрессии А (MM) по модифицированному методу Гаусса и рассчитывает коэффициент множественной корреляции RR.

Обращение к подпрограмме MISR, которая вычисляет значения зависимой переменной и значение критерия регулярности:

$$\text{REGA} = \frac{\sum\limits_{k=2}^{N2} |y_k - x_{1,k}|^2}{\sum\limits_{k=N1}^{N2} x_{1,k}^2}; \quad (3.30)$$

$$\text{REG}_1 = \frac{1}{N2 - N1 + 1} \sum\limits_{k=N1}^{N2} \left| \frac{y_k - x_{1,k}}{x_{1,k}} \right|^2; \quad (3.31)$$

$$\text{REG}_2 = \sqrt{\frac{1}{N2 - N1 + 1} \sum\limits_{k=N1}^{N2} \left| \frac{y_k - x_{1,k}}{x_{1,k}} \right|^2}; \quad (3.32)$$

$$\text{REG}_4 = \frac{1}{N2 - N1 + 1} \sum\limits_{k=N1}^{N2} \left| \frac{y_k - x_{1,k}}{(y_k + y_{1,k})/2} \right|^2; \quad (3.33)$$

$$\text{REG}_5 = \sqrt{\frac{1}{N2 - N1 + 1} \sum\limits_{k=N1}^{N2} \left| \frac{y_k - x_{1,k}}{(y_k + x_{1,k})/2} \right|^2}. \quad (3.34)$$

На печать выводятся входные параметры массив реальных данных, средние значения переменных, дисперсии первичных массив нормированных данных, массив коэффициентов парной корреляции вектор коэффициентов регрессии, значение критерия регулярности B, значение критерия, значение множественной корреляции, значение критерия регулярности В в случае отсутствия решения системы нормальных уравнений сообщение «Нет решения».

## ПРОГРАММА 35

```

1.      SUBROUTINE REGR(N1,N2,N,M,LL,RR,KGA,NN,MM,X,X1,XC,SX,Z,
2.      *B,F,R,A,Y,REG)
3.      C      РАСЧЕТ КОЭФ. ЛИНЕЙНОЙ РЕГРЕССИИ И КРИТЕРИЕВ ОЦЕНOK
4.      C      УРАВНЕНИЯ РЕГРЕССИИ
5.      REAL X(NN,MM),X1(NN,MM),XC(MM),Z(MM,MM),R(MM,MM),
6.      *F(MM),R(MM),A(MM),U(NN),REG(LL)
7.      COMMON /KEY/ KIJ
8.      100 FORMAT(//,T20,'ИСХОДНЫЕ ДАННЫЕ',//)
9.      101 FORMAT(1X,10G12.6)
10.     102 FORMAT(//,T20,'СРЕДНИЕ ЗНАЧЕНИЯ',//)
11.     103 FORMAT(//,T20,'ДИСПЕРСИИ ПЕРВЫХ',//)
12.     105 FORMAT(//,T20,'КОРРЕЛЯЦИОННЫЕ КОЭФФИЦИЕНТЫ',//)
13.     104 FORMAT(//,T20,'НОРМИРОВАННЫЕ ДАННЫЕ',//)
14.     107 FORMAT(//,T20,'КОЭФФИЦИЕНТЫ РЕГРЕССИИ')
15.     106 FORMAT(10X,5G20.10,/)
16.     108 FORMAT(//,T20,'ЗНАЧЕНИЕ УРАВНЕНИЯ РЕГРЕССИИ',//)
17.     109 FORMAT(//,T20,'ЗНАЧЕНИЯ КРИТЕРИЕВ',//)
18.     CALL REG1(N1,N2,N,M,RR,KGA,NN,MM,X,X1,XC,SX,Z,B,F,R,A)
19.     IF(KGA.EQ.0) GOTO 2
20.     CALL MISR(N1,N2,N,M,NN,MM,0,LL,X,A,Y,REG)
21.     IF(KIJ.NE.0) GOTO 1
22.     WRITE(3,100)
23.     WRITE(3,101)((X(I,J),I=1,N),J=1,M)
24.     WRITE(3,102)
25.     WRITE(3,101)(REG1,I=1,M)
26.     WRITE(3,103)
27.     ~   WRITE(3,101)(SX(I),I=1,M)
28.     WRITE(3,104)
29.     WRITE(3,104)
30.     WRITE(3,105)

```

31. WRITE(3,106)((Z(I,J), I=1,M), J=1,M)

32. WRITE(3,107)

33. WRITE(3,101)(A(I), I=1,M)

34. WRITE(3,108)

35. WRITE(3,101)(Y(I), I=1,N)

36. WRITE(3,109)

37. WRITE(3,101)(REG(I), I=1,5)

38. 1 CONTINUE

39. 2 RETURN

40. END

## ПОДПРОГРАММА NORM

Вычисление средних значений, дисперсии и нормировка данных входного массива (прогр. 36)

Обращение CALL NORM(N1, N2, N, M, MM, NN, X, X1, XC, SX)

Описание параметров Описаны на с 94

Требуемые подпрограммы SQRT.

## ПРОГРАММА 36

1. SUBROUTINE NORM(N1,N2,N,M,NN,MM,X,X1,XC,SX)

2. REAL X(NN,MM),XC(MM),SX(MM),XI(NN,MM)

3. N>=N2-N1+1

4. DO 1 J=1,M

5. XC(J)=0. 14. \*FLOAT(N3))

6. DO 2 I=N1,N2 15. DO 4 I=N1,N2

7. 2 XC(J)=XC(J)+X(I,J) 16. 4 XI(I,J)=(X(I,J)-

8. XC(J))/SX(J) 17. \*XC(J))/SX(J)

9. SX(J)=0. 18. 1 CONTINUE

10. DO 3 I=N1,N2 19. RETURN

11. 3 SX(J)=SX(J)+X(I,J)~

12. XC(J)\*\*2 20. END

13. SX(J)=SORT SX(J)/

## ПОДПРОГРАММА CORA

Вычисление матрицы парной корреляции по значениям входного массива (прогр. 37)

Обращение CALL CORA (N1, N2, N, M, NN, MM, X, Z)

Описание параметров Описаны на с 94

Требуемые подпрограммы Нет

1. SUBROUTINE CORA(N1,N2,N,M,NN,MM,X,Z) ПРОГРАММА 37

2. REAL X(NN,MM),Z(MM,MM)

3. DO 5 I=1,M

4. DO 5 I=1,M 13. GOTO 5

5. IF(J-I)>7,9,8 14. 8 IF(J.GT.1) Z(I,J)=

6. 7 Z(I,J)=0. 15. \*Z(J,I)

7. N2=N2-N1+1 16. GOTO 5

8. DO 6 K=N1,N2 17. 9 IF(J.EQ.1) Z(I,J)=1

9. 6 Z(I,J)=Z(I,J)+ 18. 5 CONTINUE

10. \*X(K,J)\*X(K,I) 19. RETURN

11. Z(I,J)=Z(I,J)/ 20. END

## ПОДПРОГРАММА REGA

Решение системы нормальных уравнений по модифицированному методу Гаусса (прогр. 38)

Описание e. CALL REGA (M, MM, XC, SX, Z, B, F, R, A, RR, KGA).

Описание параметров Описаны на с 94

Требуемые подпрограммы SQRT.

Метод Первый переносится считается зависимой оставляемые ( $M - 1$ ) — независимыми переменными. На основе матрицы коэффициентов парной корреляции  $Z_{(MM)}$  по модифицированному методу Гаусса вычисляется вектор коэффициентов  $A_{(M)}$  в уравнении линейной регрессии

## ПРОГРАММА 38

1. SUBROUTINE REGA (M,MM, XC, SX, Z, B, F, R, A, RR, KGA)

2. REAL Z(MM,MM),B(MM,MM),F(MM),R(MM),A(MM),

3. \*SX(MM),XC(MM)

4. KGA=1

5. DO 11 I=2,M 15. DO 14 I=K,MM

6. F(I-1)=Z(1,I) 16. DO 14 J=K,MM

7. DO 11 J=2,M 17. IF(Abs B(I,J)).LE.

8. 11 B(I-1,J-1)=Z(I,J) 18. \*SM) GOTO 14

9. M1=M-1 19. SM=AES(B(I,J))

10. DO 12 K=1,MM 20. IM=I

11. 12 R(K)=K 21. JM=J

12. C 22. 14 CONTINUE

13. 10 13 K=1,MM 23. IF(SM.EQ.0) GOTO 16

14. SM=0. 24. IF(IM.EQ.J) GOTO 101

25. DO 17 I=K,M1

58.C

26. SM=B(K,I)

59.

27. B(K,I)=B(M,I)

60.

28. B(IM,I)=SM

61.

29. 17 CONTINUE

62.

30. DO 23 I=K1,M1

63.

31. IF(R)=F(K)

64.

32. F(1M)=SM

65.

33. 101 IF(JM.EQ.K) GOTO 102

66.

34. DO 18 I=1,M1

67.

35. SM=B(I,K)

68.

36. B(I,K)=B(I,JM)

69.

37. B(I,JM)=SM

70.

38. 18 CONTINUE

71.

39. 102 SM=1/B(K,K)

72.

40. R(K)=R(JM)

73.

41. R(JM)=SM

74.

42. 103 SM=R(K)

75.

43. K1=K+1

76.

44. DO 19 I=K1,M1

77.

45. 19 B(K,I)=B(K,I)+SM

78.

46. F(K)=F(K)\*SM

79.

47. DO 20 I=1,M1

80.

48. IF(I.EQ.K) GOTO 103

81.

49. SM=B(I,K)

82.

50. K1=K+1

83.

51. DO 21 J=K1,M1

84.

52. 21 B(I,J)=B(I,J)-B(K,J)\*85.

85.

53. \*SM

86.

54. F(I)=F(I)-F(K)\*SM

87.

55. 103 CONTINUE

88.

56. 20 CONTINUE

89.

57. 13 CONTINUE

90.

58. END

## ПОДПРОГРАММА MISR

Вычисление значений критерия регулярности (прогр. 39)

Обращение CALL MISR (N1, N2, N, M, NN, MM, LL, X, A, Y, REG)

Описание параметров:

У (NN) — вспомогательный массив. Остальные параметры описаны на с 94.

Требуемые подпрограммы SORT.

### ПРОГРАММА 39

```
SUBROUTINE MISR(N1,N2,N,M,NN,MM,LL,X,A,Y,REG)
REAL X(NN,MM),A(MM),Y(NN),REG(LL)
```

```
1.      SUBROUTINE MISR(N1,N2,N,M,NN,MM,LL,X,A,Y,REG)
2.      REAL X(NN,MM),A(MM),Y(NN),REG(LL)
```

```
3.      N3=N2+N1+1
4.      DO 1  I=1,N
```

```
5.      Y(K)=A(1)
6.      DO 1  I=2,M
```

```
7.      1  Y(K)=Y(K)+A(1)*X(K,1) 22.
8.      C1=0.
9.      C2=0.
```

```
10.     DO 2  I=2,5
11.     2  REC(1)=0.
12.     DO 3  K=N1,N2
13.     C4=Y(K)*X(K,1)
14.     C5=Y(K)*X(K,1)
15.     C1=C1+ABS(C4)**2
16.     C2=C2*X(K,1)**2
17.     REC(2)=REC(2)+ABS(
18.     *C4/X(K,1))
19.     REC(3)=REC(3)+(ABS(
```

```
20.     *C4/X(K,1)) 33.
21.     REC(1)=C1/C2
22.     REC(1)=SORT(REC(1))
23.     REC(5)=REC(5)+ABS(
24.     REC(4)=REC(4)**2
25.     REC(4)=REC(4)**2
26.     REC(1)=REC(1)/FLOAT(N3))
27.     REC(1)=C1/C2
28.     REC(1)=SORT(REC(3))
29.     REC(3)=SORT(REC(3))
30.     REC(4)=REC(4)**2
31.     REC(5)=2*SQRT(REC(5))
32.     RETURN
33.     END
```

## ПОДПРОГРАММА DICH

Перевод десятичного числа в число с заланным основанием (прогр. 40).

Обращение CALL DICH, (JQ, ID, JN, JS)

Описание параметров:

JQ — заданное целое десятичное число,

JS — основание выбранной системы исчисления;

ID (JN) — вектор, элементами которого являются разряды переведенного числа.

Преобразуемые подпрограммы Нет

### ПРОГРАММА 40

```
SUBROUTINE DICH(JQ, ID, JN, JS)
DIMENSION ID(JN)
```

```
1.      SUBROUTINE DICH(JQ, ID, JN, JS)
2.      DIMENSION ID(JN)
```

```
3.      JI=10
4.      DO 11  I=1,JN
```

<p><b>5.</b>      <b>11</b> ID(I)=0</p> <p><b>6.</b>      <b>12.</b>    ID(JN1-1)=JL-JC*JS</p> <p><b>7.</b>      <b>13.</b>    JL=JC</p> <p><b>8.</b>      <b>14.</b>    GOTO 16</p> <p><b>9.</b>      <b>15.</b>    18 ID(JN1-1)=JL</p> <p><b>10.</b>     <b>16.</b>    19 RETURN</p> <p><b>11.</b>     <b>17.</b> JC=JL/J</p> <p><b>18.</b>     <b>END</b></p>
<p><b>ПОДПРОГРАММА МАН1</b></p>
<p>Нахождение первого максимального элемента данного вектора (прогр. 41).</p> <p><b>Обращение.</b> CALL МАН1 (G, JE, C, M).</p> <p><b>Описание параметров:</b></p> <ul style="list-style-type: none"> <li>G (JE) — входной вектор;</li> <li>JE — число элементов входного вектора;</li> <li>C — значение максимального элемента входного вектора;</li> <li>M — номер первого элемента с максимальным значением.</li> </ul> <p><b>Требуемые подпрограммы.</b> Нет.</p>
<p><b>ПРОГРАММА 41</b></p>
<ol style="list-style-type: none"> <li>1. SUBROUTINE МАН1(G,JE,C,M)</li> <li>2. DIMENSION G(JE)</li> <li>3. C=G(1)</li> <li>4. M=1</li> <li>5. I=2</li> <li>6. 20 IF(G&lt;G(I)) 21,22,22</li> <li>7. 21 G=G(I)</li> <li>8. M=I</li> <li>9. 22 I=I+1</li> <li>10. 1F(I-JE) 20,20,23</li> <li>11. 23 RETURN</li> <li>13. END</li> </ol>
<p><b>ПОДПРОГРАММА RAZR</b></p>
<p>Перераспределение элементов вектора A в зависимости от значений элементов бинарного вектора ID (прогр. 42).</p> <p><b>Обращение.</b> CALL RAZR (MV, M, M4, M41, ID, A).</p> <p><b>Описание параметров:</b></p> <ul style="list-style-type: none"> <li>M — максимальная длина вектора A (M);</li> <li>M4 — длина бинарного вектора ID (M41);</li> <li>M4 — номер последнего элемента вектора A (M), до которого он преобразуется;</li> <li>MV — число элементов вектора A (M), которое необходимо переставить согласно значениям вектора ID (M41);</li> <li>ID (M41) — выходной бинарный вектор;</li> <li>A (M) — вектор значений, которые нужно переставить.</li> </ul> <p><b>Требуемые подпрограммы.</b> Нет.</p> <p><b>Метод.</b> Первые M элементы вектора A (M) переставливаются последовательно по следующему правилу: если ID (I) = 0, то A (I+1) = 0; при ID (I) = 1 A (I+1) = A (MV) и MV = MV - 1, где I = (M4 - 1), ..., 1.</p>

ПОДПРОГРАММА МАН

**Нахождение первого максимального элемента данного вектора (прогр. 41).**  
*Обращение. CALL МАН1 (G, JE, C, M).*  
*Описание параметров:*

**G (JE)** — число элементов входного вектора;

**C** — значение максимального элемента входного вектора;

**M** — номер первого элемента с максимальным значением.

*Требование подпрограммы. Нет.*

ПОДПРОГРАММА RAZR

**Переопределение элементов вектора A в зависимости от значений элементов бинарного вектора ID (прогр. 42).**

**Обращение. CALL RAZR (MV, M, M4, M41, ID, A).**

**Описание параметров:**

- M** — максимальная длина вектора  $A(M)$ ;
- M4** — длина бинарного вектора  $ID(M4)$ ;
- M4** — номер последнего элемента вектора  $A(M)$ , до которого он преобразуется;
- MV** — число элементов вектора  $A(M)$ , которое необходимо переставить согласно значениям вектора  $ID(M4)$ ;
- ID (M41)** — выходной бинарный вектор;
- $A(M)$  — вектор значений, которые нужно переставить.

*Требуемые подпрограммы. Нет.*

*Метод.* Первые  $M4$  элементов вектора  $A(M)$  перестанавливаются последовательно по следующему правилу: если  $ID(I) = 0$ , то  $A(I+1) = 0$ ; при  $ID(I) = 1$   $A(I+1) = A(MV)$  и  $MV = MV - 1$ , где  $I = (M4 - 1), \dots, 1$ .

ПОДПРОГРАММА ZAR

**Заполнение массива  $X$  (NN, M4) линиями из массива  $Z_N$  (NN, MM) в зависимости от значений элементов блочного пектора  $ID$  (M41) [программа 43].**  
**Обращение:** CALL ZAR (NN, NN, MM, M41, ZN, X, ID, IIP, IQ, IQQ, M).  
**Описание параметров:** Описаны на с. 94.  
**Преимущества подпрограммы:** DICH.  
**Метод.** Обращение к подпрограмме DICH для заполнения вектора  $ID$  (M41). В зависимости от значений элементов вектора  $ID$  (M41) заполняется массив  $X$  (NN, M4) по следующему правилу:

данного вектора (прогр. 41).

если  $ID(1) = 1$ , то  $M = M + 1$  и  $X(T, M) = ZN(T)$   
 если  $ID(2) = 1$ , то  $M = M + 1$  и  $X(T, M) = ZN(T)$

**M41** — длина бинарного вектора ID (M41);

**M4** — номер последнего элемента вектора A (M), до которого он преобразуется;

**MV** — согласно значениям вектора ID (M41);

**ID (M41)** — выходной бинарный вектор;

**A (M)** — вектор значений, которые нужно переставить.

*Требуемые подпрограммы.* Нет.

*Метод.* Первые M4 элементов вектора A (M) переставливаются последовательно по следующему правилу: если  $ID (I) = 0$ , то  $A (I+1) = 0$ ; при  $ID (I) = 1$   $A (I+1) = A (MV)$  и  $MV = MV - 1$ , где  $I = (M4 - 1), \dots, 1$ .

ПРОГРАММА 42

## ПОДПРОГРАММА KRIT

Вычисление комбинированного критерия (прогр. 44).

*Обращение.* CALL KRIT (LL, REG).

*Описание параметров:* Описаны на с. 94.

*Требуемые подпрограммы.* Нет.

**REG<sub>1</sub> + REG<sub>7</sub>**

### ПРОГРАММА 44

```

1. SUBROUTINE KRIT(LL,REG)
2. REAL REG(LL)
3. REG(8)=REG(1)*REG(6)
4. REG(9)=REG(1)*REG(7)
5. RETURN
6. END

```

Вычисление значения критерия несмешенности (прогр. 45).

*Обращение.* CALL MISN (N, M, NN, MM, LL, X, AA, AB, AAB, YA,

UV, UAB, REG).

*Описанные параметры:* Описаны на с. 94.

*Требуемые подпрограммы.* Нет

### ПОДПРОГРАММА MISN

```

1. SUBROUTINE MISN(N,M,NN,MM,LL,X,AA,AB,AAB,YA,
2. UV,UAB,REG)
3. *REG(LL)
4. D2=0.
5. D3=0.
6. D4=0.
7. DO 1 K=1,LL
8. YAK(K)=AA(K)
9. YBK(K)=AB(K)
10. YAB(K)=AAB(K)
11. DO 2 I=2,M
12. YAI(K)=YAI(K)+AA(I)*X(K,I) 23.
13. YBI(K)=YBI(K)+AB(I)*Y(I,K) 24.
14. YAB(K)=YAB(K)+AAB(I)*Y(I,K)
15. *AA(K)=YAK(K)
      25. END

```

## ПОДПРОГРАММА EKSTR

Вычисление средних, минимальных и максимальных значений входного вектора (прогр. 46)

*Обращение.* CALL EKSTR (NV, SMX, SMN, SSR, IMX, IMN, MNU, S).

*Описание параметров:* Описаны на с. 94.

*Требуемое подпрограммы.* FLOAT

### ПРОГРАММА 46

```

1. SUBROUTINE EKSTR(S,NV,SMN,SMX,SSR,IMN,IMX)
2. REAL S(NV)
3. SSR=S1
4. SMX=S1
5. SMN=S1
6. DO 1 I=2,NV
7. IF(SMX.LT.S(I)) SMX 15.
8. *S1
9. IF(SMN.GT.S(I)) SMN 17.
10. *S(I)
11. IF(SMN.GT.S(I)) IMN 18.
12. *S1
13. IF(SMN.GT.S(I)) SMN 17.
14. *S(I)
15. SSR=SSR+S(I)
16. 1 CONTINUE
17. RETURN

```

## Глава 4

### МОДЕЛИРОВАНИЕ ПОЛЕЙ

#### 1. ЗАДАЧА МОДЕЛИРОВАНИЯ ФИЗИЧЕСКИХ ПОЛЕЙ

При моделировании физических полей идентифицируются (строятся) картины поля и оцениваются его параметры. Математические модели поля необходимы для прогнозирования его дальнейшего движения как во времени, так и в пространстве, а также для синтеза закона оптимального управления полем. При мером физического поля, подлежащего моделированию, может служить поле загрязнения воздушного или водного бассейна, поле ядерного реактора и т. д.

В тех случаях, когда количество исходных данных мало, практически исключена возможность применения каких-либо вероятностных моделей, основанных на построении многомерных распределений. Реальная возможность моделирования связана с применением методов самоорганизации моделей на ЭВМ, основанных на перевороте большого числа моделей — претендентов (в форме конечно-разностных уравнений) на ЭВМ. Эти методы требуют сравнительно небольшого числа точек измерений, и при выборе внешнего критерия селекции соответствующего вида обладают значительной помехоустойчивостью (см. гл. 2).

Для построения поля используются данные наблюдений в точках поля, данные о месторасположении, интенсивности и времени действия возмущающих воздействий. В задаче моделирования поля загрязнения воздушного и водного бассейнов — это соответственно данные наблюдений контрольных станций и данные о месторасположении, интенсивности и времени выборок загрязняющих веществ. Выбор выходной величины и аргументов определяет постановку задачи моделирования, которая зависит от цели последнего (интерполяция, экстраполяция или прогноз), а также от наличия исходной информации, используемой для синтеза модели.

В лаптой глянсе описаны программы моделирования полей, которые применимы для решения задачи нахождения закона и величины в сле- дующих постановках: по данным измерений в некото- рых точках поля; по инфор- мации о возмущении; по тем и другим данным вмес- те

форме) не задается априори, а находится на основании опытных данных по прин- ципу самоорганизации при помощи перебора шаблонов и величинностей. При этом число членов полного уравнения обычно значительно превышает число опытных данных. Оценки коэффициентов находятся по методу наименьших квадратов; ре- шение — при помощи пошаго- вого интегрирования конечно- разностного уравнения.

Во второй постановке аргу- менты конечно-разностной мо- дели выбираются так, чтобы вид уравнений соответствовал матрице вход — выход.

Все зависимости от поста- новки задачи моделирования об- ластию интерполяции на всем ее протяжении, лежащую внутри многогранника, вершина- ми которого являются погра- ничные станицы. Об

ласть экстраполяции лежит вне этого (в общем случае) трехмер- ного многогранника (рис. 7). На рис. 7 показаны шаблоны для экстраполяции в направле- нии  $x$  области загрязнения, заключенной между контрольными стан- циями 1, 2 и 3

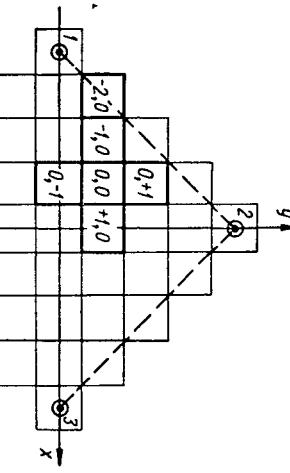


Рис. 7. Применение шаблонов для экстраполяции в направлении оси загрязнения, заключенной между контрольными станциями 1, 2, и 3

При прогнозировании область интерполяции лежит внутри временного интер- вала опытных данных. Область прогнозирования лежит вне области интерполяции процесса. Характерной особенностью области интерполяции является то, что, согласно теореме Бейерштрасса логарифмическая кривая, описываемая достаточно сложной функцией, укладывается среди опытных точек достаточно хорошо. Только в об- ласти экстраполяции и прогноза кривые быстро расходятся, образуя так называемый «веер» прогнозов. Модель должна соответствовать той функции (или решению дифференциального уравнения), которая в области экстраполяции или прогнозиро- вания должна лучше других соответствовать будущему ходу процесса.

В зависимости от принципа выбора аргументов можно различить три задачи первой задачи основана на принципе «непрерывности» или «ближнего действия». Вторая — на противоположном принципе «дальнего действия», в третьей задаче, комбинированной, используются «одновременно» принципы «дальнего и ближ- него действия».

Модель может быть точечной или пространственной, одномерной, двумерной, трехмерной или многомерной, алгебраической, гармонической или конечно-раз-ностной (дифференциальной). Если модель строится по данным наблюдений, в ко- торых не указано пространственное размещение датчиков (или контрольных стан- ций) то она называется *точечной*. В этом случае задача решается так, как будто все данные собраны в одной точке пространства.

Модель называется *пространственной*, или моделью поля, если в исходных данных содержится информация о размещении контрольно-измерительных стан- ций в пространстве. Пространственные модели требуют наличия не менее трех станций по каждой оси. Модели с одним аргументом называются одномерными, а с несколькими — многомерными.

**Первая задача.** Моделирование поля по конечно-разностным аналогам урав- нений турбулентной диффузии, основанных на принципе «ближнего действия». Уравнения турбулентной диффузии основаны на принципе «непрерывности» или «ближнего действия» соседних друг другу частиц, образующих физическое поле.

Элементарным шаблоном в теории конечно-разностных уравнений называется геометрическая фигура, показывающая, какие именно соседние точки поля исполь- зуются для построения структуры уравнения. Изменения поля в двух соседних клетках (кубиках) достаточно для отображения первой производной (по оси распо- ложения ячеек), а в трех клетках — для аналого второй производной [24].

Примеры соответствия линейных дифференциальных уравнений их конечно-разностным аналогам и элементарным шаблоном показаны на рис. 8 и в табл. 7.

Для физических полей обычно известны некоторые детерминированные модели поля, математически выражаемые в виде дифференциальных или интегро-дифференциальных уравнений. Такие уравнения можно использовать в качестве образца при выборе вида аргументов и функций, участвующих в полном описании модели (комбинированный метод). Детерминированное уравнение диффузии ука- зывает на определенный шаблон, который целесообразно принять за основу пере-

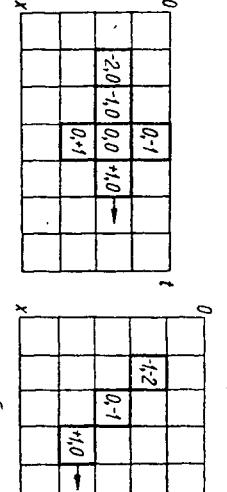


Рис. 8. Поля интерполяции цифровых данных типа «самолет» (a) и «диагональ» (b)

с поочередным прививанием шаблона, то есть или иных коэффициентов полного полинома и многорядный (поготовый).

**Перебор шаблонов и перебор степени величинности («шаблонной перебор»).**

При числе слагаемых полного полинома меньше двадцати для выбора модели оп- тимальной сложности используется комбинаторный алгоритм МГУА (с зануле- нием всевозможных комбинаций членов полного полинома), предусматривающий перебор всех возможных частных полиномов. В этом случае достаточно рассмотреть только один, наиболее полный, заметно достаточный шаблон, степень кото- рого равна числу аргументов; тогда перебор шаблонов не требуется. При числе слагаемых полного полинома больше 20 для выбора модели используется много- рядный (поготовый) алгоритм МГУА, при котором возможна потеря некоторых частных полиномов.

Чтобы расширить перебор, целесообразно применить: перебор шаблонов (со- става аргументов), начиная с самого простого двухячейочного шаблона (рис. 8, б); перебор всевозможных полиномов для каждого из шаблонов по многорядному ал- горитму МГУА с перебором аргументом переносных, начиная с полиномом первого степеня и кончая полиномом, в котором высшая степень равна числу аргументов, указанному в данном шаблоне.

Такой двойной перебор шаблонов и величинных полиномов дает возможность более полно пересмотреть множество возможных частных полиномов в задачах, где из-за большого числа аргументов нельзя применить более простой комби- наторный алгоритм МГУА.

При решении первой задачи применяется следующая постановка: *Моделирова-ние поля загрязнения по данным нескольких точек измерения (первая задача, первая постановка  $I = 1$ )*.

Первая постановка задачи моделирования поля заключается в построении поля не только в области интерполяции, но и на значительном расстоянии за пределами этой области, т. е. в экстраполяции поля и прогнозе его развития во врем- мени. При этом предполагается, что возмущения сравнительно мало изменяются во времени, поэтому непосредственно не учитываются. Таким образом, в первой постановке множество аргументов содержит в себе только данные о параметрах поля в определенных точках. Если таких точек мало, то (как указывалось) внутрь

области интерполяции данные можно интерполировать, чтобы получить средние значения показателей параметров во всех квадратах поля  $t - x$ . Двухэтапная интерполяция [22] состоит в том, что сначала данные измерений интерполируются (между точками) по методу наименьших квадратов. Полученное «цифровое поле» используется для спектра количественно-распределительных моделей, пригодных для прогнозирования.

Ниже показаны уравнения только для двумерной модели  $q = f(x, t)$ . По аналогии можно составить подобные уравнения и для трехмерного (или четырехмерного) поля.

Уравнение для параметров поля при шаблоне 5 (см. табл. 7) можно записать в векторной форме:

$$q_{i(+1,0)}^* = \alpha_{i+1} [q_{k(0,0)}, q_{k(0,1)}, q_{k(0,+1)}, q_{k(-1,0)}, q_{k(-2,0)}, \dots, q_{k(-\tau,0)}]; \quad (4.1)$$

для экстраполяции

$$q_{i(-1,0)} = \alpha_{i-1} [q_{k(0,0)}, q_{k(-1,0)}, q_{k(+1,0)}, q_{k(0,-1)}, q_{k(0,-2)}, \dots, q_{k(0,-\tau)}],$$

где  $\alpha_j$  — вектор полиномиальных функций (линейные, нелинейные, без ковариаций или с неполным учетом ковариаций) размерности  $j$ .

Во всех приводимых уравнениях вектор измеряемых внешних переменных, стоящий в правой части, в общем случае может содержать как сами переменные, так и их запаздывающие значения, а также ковариации (по две), которые можно рассматривать как отдельные переменные, например,

$$q_{(0,0)}, q_{(2,0)}, \dots, q_{(t-1)}, q_{(t-1)}, \dots, (q_{(t,0)}, q_{(2,0)}), (q_{(t,0)}, q_{(t-1)}) \dots$$

В рассмотренной двумерной задаче не учитывался процесс диффузии компонент поля из слоя и оседания частиц по вертикали. Для учета этих явлений вводится так называемая «функция источника» или «остаток» — функция координат квадрата, в котором измеряется выходная величина. Кроме того, в функцию источника входит возмущающее воздействия (учет оседания частиц за прязняющих веществ и боковой пригодности загрязнений, а также внешние воздействия среды).

**Вторая задача: Моделирование полей по конечно-разностным уравнениям, соответствующим расширенной матрице «вход — выход» (основанной на принципе «дальнего действия»).**

Кроме аналогов уравнений турбулентной диффузии, для моделирования физических полей применяются уравнения, соответствующие функциональной матрице «вход — выход».

Решение второй задачи моделирования поля удобно, если требуется дать прогноз изменения поля внутри области интерполяции при изменении возмущений во времени. Для экстраполяции поля далеко за пределы указанной области решение второй задачи не обеспечивает точности. Вторая задача имеет три постановки. Первая постановка (II-1): Моделирование по матрице «вход — выход», использующее информацию исключительно точек измерения. В первой постановке для  $i$ -й точки используется следующие уравнения прогноза:

$$q_{i(+1)}^f = \sum_{S=1}^m \alpha_{Si} (q^S), \quad (4.2)$$

где

$$q^S = [q_{i(0)}^S, q_{i(-1)}^S, \dots, q_{i(-\tau)}^S, \dots, q_{i(0)}^S, q_{i(-1)}^S, \dots, q_{i(-\tau)}^S], \quad i = 1, n,$$

$q^S$  — параметры поля;  $\alpha$  — оператор полиномиальных функций;  $m$  — количество точек измерения;  $n$  — количество компонент. Например, при  $m = 3, n = 2, \tau = 2$

$$q_{i(+1)}^f = \alpha_{21} [q_{i(0)}^{(2)}, q_{i(-1)}^{(2)}, q_{i(-2)}^{(2)}, q_{i(0)}^{(2)}, q_{i(-1)}^{(2)}, q_{i(-2)}^{(2)}] + \alpha_{31} [q_{i(0)}^{(3)}, q_{i(-1)}^{(3)}, q_{i(-2)}^{(3)}, q_{i(0)}^{(3)}, q_{i(-1)}^{(3)}, q_{i(-2)}^{(3)}]. \quad (4.3)$$

Таблица 7  
Перебор шаблонов формы диагонали и получение данных для двумерной задачи прогнозирования

Шаблоны	Таблицы данных	Поступление модели и аналого
		$\frac{\partial q}{\partial t} + a_1 q = f(t, x)$ $+ f(q_{0,0})$
		$\frac{\partial q}{\partial t} + a_1 \frac{\partial q}{\partial x} + a_2 q =$ $= f(t, x)$ $q_{i+1,0} = f_1(t_{i+1,0}, x_{i+1,0}) +$ $+ f_2(q_{0,0}, q_{0,-1})$
		$\frac{\partial q}{\partial t} + a_1 \frac{\partial^2 q}{\partial x^2} + a_2 q +$ $+ a_3 \frac{\partial q}{\partial x} + a_4 q = f(t, x)$ $q_{i+1,0} = f_1(t_{i+1,0}, x_{i+1,0}) +$ $+ f_2(q_{0,0}, q_{0,-1})$
		$\frac{\partial^3 q}{\partial t^3} + a_1 \frac{\partial^2 q}{\partial t^2} +$ $+ a_2 \frac{\partial^2 q}{\partial x^2} + a_3 \frac{\partial q}{\partial t} +$ $+ a_4 \frac{\partial q}{\partial x} + a_5 q = f(t, x)$ $q_{i+1,0} = f_1(t_{i+1,0}, x_{i+1,0}) +$ $+ f_2(q_{0,0}, q_{0,-1},$ $q_{0,-1}, q_{-1,0}, q_{-1,-1})$

Продолжение табл. 7

Шаблоны	Габлины данных	Постепенное сложение модели и аналога
		$\frac{\partial q}{\partial t} + a_1 \frac{\partial q}{\partial x} + a_2 q = f_1(t_{i+1,0}, x_{i+1,0}) + f_2(q_{i-1,0})$
		$\frac{\partial^2 q}{\partial t^2} + a_1 \frac{\partial^2 q}{\partial x^2} + a_2 \frac{\partial q}{\partial t} + a_3 \frac{\partial q}{\partial x} + a_4 q = f_1(t_{i,0}, x_i) + f_2(q_{i-1,0})$
		$\frac{\partial^3 q}{\partial t^3} + a_1 \frac{\partial^3 q}{\partial x^3} + a_2 \frac{\partial^2 q}{\partial t^2} + a_3 \frac{\partial^2 q}{\partial x^2} + a_4 \frac{\partial q}{\partial t} + a_5 \frac{\partial q}{\partial x} + a_6 q = f_1(t_{i+1,0}, x_{i+1,0}) + f_2(q_{i-1,0}, q_{i-2,0})$
		$\frac{\partial^4 q}{\partial t^4} + a_1 \frac{\partial^4 q}{\partial x^4} + a_2 \frac{\partial^3 q}{\partial t^3} + a_3 \frac{\partial^3 q}{\partial x^3} + a_4 \frac{\partial^2 q}{\partial t^2} + a_5 \frac{\partial^2 q}{\partial x^2} + a_6 \frac{\partial q}{\partial t} + a_7 q = f_1(t_{i,0}, x_i) + f_2(q_{i-1,0}, q_{i-2,0})$

$$q_{i(+1)}^l = \beta_{11} [\lambda_{1(0)}^{(1)}, \lambda_{1(-1)}^{(1)}, \lambda_{1(-2)}^{(1)}, \lambda_{2(0)}^{(1)}, \lambda_{2(-1)}^{(1)}, \lambda_{2(-2)}^{(1)}] + \beta_{21} [\lambda_{1(0)}^{(2)}, \lambda_{1(-1)}^{(2)}, \lambda_{1(-2)}^{(2)}, \lambda_{2(0)}^{(2)}, \lambda_{2(-1)}^{(2)}, \lambda_{2(-2)}^{(2)}]. \quad (4.5)$$

Например, при  $n = 2, r = 2, p = 2$  будет

Третья постановка (II-3): Моделирование, учитывающее информацию о таком изменении и о возмущениях: В общей (третьей) постановке задачи матрица «вход — выход» расширяется вдвое в нее наряду с возмущениями данных «зарядов» в ближайших точках

В общей постановке для  $i$ -й точки уравнение прогноза

$$q_{i(+1)}^l = \sum_{S=1}^m \alpha_{Si} (q^S) + \sum_{S=1}^p \beta_{Si} (\lambda^S). \quad (4.6)$$

В матрице участвуют как возмущения  $\lambda$ , так и параметры поля  $q$ . «Функция источника». Для учета внешних воздействий (приведенного ветра или течения, температуры вытяжности и других физических величин) в уравнения (4.2), (4.4), (4.6) рекомендуется добавить в качестве слагаемого правой части функцию источника

Третья задача: Моделирование полей при одновременном использовании принципов «ближнего» и «дальнего действия». Эта задача имеет три постановки:

1. Моделирование, использующее принцип «ближнего действия» и матрицу «вход — выход», учитывающую сочетание в точках измерения (постановка III-1). В этой постановке используются сочетания постапавок I-1 и II-1.
2. Моделирование, использующее принцип «ближнего действия» и матрицу «вход — выход», учитывающую информацию о возмущениях (вторая постановка III-2), что является комбинацией постановок I-1 и II-2.
3. Моделирование, использующее принцип «ближнего действия» и расширенную матрицу «вход — выход», учитывающую как правило в точках измерения, так и о возмущениях (постановка III-3). В этой постановке используется сочетание подходов к решению постановок I-1 и II-3.

## 2. ПРОГРАММА МГУ ДЛЯ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ ПРОЦЕССОВ

Предназначена для построения математических моделей пространственно-временных процессов в виде конечно-разностных уравнений с учетом запаздывающих аргументов по результатам патрульных наблюдений (прогр. 47).

Язык программирования АЛГОЛ, БЭСМ-6, Используемые подпрограммы: FORM, GAUSS, MAX, NORM, RANK, NM.

ОП, SUM.

Метод. Необходимо найти одну из зависимостей ((4.1), (4.2), (4.4), (4.6)) выходной величины от входных аргументов в виде конечно-разностных уравнений, при этом используется один из шаблонов (см. габл. 7). С помощью преобразования переменных пятичленной в общем случае описание приводится к линейному виду и в дальнейшем на каждом ряде селекции используется только линейная базовая функция (линейное частное описание). Например, процесс описывается полным полиномом второго порядка вида

$$q_{i+1,0} = a_1 q_{0,0} + a_2 q_{-1,0} + a_3 q_{-2,0} + a_4 q_{0,0} q_{-1,0} + a_5 q_{0,0} q_{-2,0} + a_6 q_{-1,0} q_{-2,0} + a_7 q_{0,0}^2 + a_8 q_{-1,0}^2 + a_9 q_{-2,0}^2.$$

Переобразованная переменные  $q_{i+1,0} = q$ ,  $q_{0,0} = W_1$ ,  $q_{-1,0} = W_2$ ,  $q_{-2,0} = W_3$ , ...  $q_{-2,0}^2 = W_9$ , получим

$$q = a_1 W_1 + a_2 W_2 + a_3 W_3 + \dots + a_9 W_9.$$

где  $q$  — параметры поля;  $\bar{W}^S$  — вектор возмущений;

$$\bar{\lambda}^S = [\lambda_{1(0)}^S, \lambda_{1(-1)}^S, \dots, \lambda_{1(-r)}^S, \dots, \lambda_{n(0)}^S, \lambda_{n(-1)}^S, \dots, \lambda_{n(-r)}^S];$$

$\beta_{Si}$  — оператор полиномальных функций;  $p$  — количество источников возмущений.

На каждом ряде селекции используется частное описание вида

$$q = b_0 + b_1 W_1 + b_2 W_2.$$

Задача определения моделей оптимальной сложности в виде количественно-разностного уравнения разбивается на два этапа. На первом этапе синтезируются и отбираются по комбинированному критерию (2.24) конечно-разностные уравнения. При этом вся выборка разбивается на части:  $A$  и  $B$ , содержащие по 40% всех точек, используемых при синтезе, — для определения среднеквадратической ошибки для отбора самых устойчивых моделей.

Коэффициенты лучших моделей, отобранных по критерию  $K_3$ , пересчитываются и отбираются по всей выборке. На втором этапе из  $F$  лучших моделей, полученных на первом этапе, окончательно отбирается одна или несколько самых устойчивых моделей по комбинированному критерию.

**Описание параметров.** *Входные:*

- $M$  — число выделенных величин, для которых синтезируются модели;
- $\Pi$  — число точек в последовательности  $A$  и  $B$  (0,6 общего числа точек выборки  $A + B + C$ , что соответствует  $A = B = 40\%$ ,  $C = 20\%$ );
- $PM$  — максимальное число рядов селекции;
- $DTAV$  — число запаздывающих аргументов;
- $17$  — число строк поля исходных данных;
- $18$  — число столбцов поля исходных данных;
- $10$  — номер экзаменационной строки, используемой для проверки моделей. При проверке по всем строкам поля  $10 = 0$ ,
- $NI$  — число точек прототипа или экстраполяции, на которых проверяется устойчивость моделей;
- $FF$  — число лучших моделей, отобранных для проверки по интегральному критерию;
- $CHO$  — свобода выбора по рядам. Количество чисел равно  $PM+1$ ;
- $DXI : IT, I : 18$  — массив исходных данных.

*Вычисляемые:*

- $M1$  — число строк массива аргументов после нелинейных преобразований;
- $NI$  — число столбцов массива аргументов после нелинейных преобразований, в выборке  $A + B + C$ ;
- $DDX [I : 17, I : 18]$  — вспомогательный массив, используемый в процедуре пошагового интегрирования;
- $Y[1 : M, I : NI]$  — массив выходной величины;
- $X[1 : M1, I : NI]$  — массив преобразованных переменных.

В случае большого количества исходных преобразованных переменных необходимо ввести описание исходных массивов в строку 7 (массивы  $DDX$  и  $DX$ ). Соответственно изменить подпрограммы FORM, а также параметры  $M1$ ,  $NI$ .

## ПОДПРОГРАММА NM

Формирование матрицы нормальных уравнений для каждого из частного описания из заготовленных заранее элементов (см. прогр. 47).

*Обращение NM (HM, XH, T)*

*Описание параметров:*

$HM$  — матрица нормальных уравнений частной модели;

$XH$  — матрица элементов нормальных уравнений всех моделей;

$T$  — параметр, кодирующий номер выборки.

*Требуемые функции и подпрограммы:* Нет.

## ПОДПРОГРАММА ОП

Формирование точек обучающей последовательности (см. прогр. 47).

*Обращение.* ОП (NP, NO).

*Описание параметров:*

NP, NO — номера точек проверочной и обучающей последовательностей соответствующим.

*Продуцент функции и подпрограммы.* Нет.

*Метод.* По заданным номерам точек проверочной последовательности формируются точки обучающей последовательности

## ПОДПРОГРАММА FORM

Формирование матрицы аргументов  $X$  и вектора выходов моделей  $Y$  (см. прогр. 47).

*Обращение* FORM (Y, X, TAУ, DX, 11, 12, 13).

*Описание параметров:*

$Y$  — выходная величина;

$U$  — числовая величина;

$X$  — матрица преобразованных входных аргументов;

$DX$  — матрица исходных матриц;

$11, 12$  — размеры исходных матриц;

$13$  — идентификатор, задающий экзаменационную последовательность;

$TAУ$  — число запаздывающих аргументов.

*Примечания:* 1. В подпрограмме предусмотрены нелинейные преобразования над исходными переменными.

2. Подпрограмма FORM составляется пользователем для конкретного шаблона.

## ПРОГРАММА 47

```

1. 'ALGO1':FORM(GAUSS,MK,NSM,RAKE,SIM.,  

2. 'ВСТИ',INTEGER'Н,М,М1,П,РМ,М1,И,Ж,К,С,Р,МТ.,'REAL'XS,XH,  

3. READM,П,РМ.),'PRINT','(10)ПР КАРТА=,Э14,//'),M,И,РМ.,  

4. 'ДЕСН',INTEGER'ARRAY'ЧС/0:М/,,,'INTEGER'И7,18,10,DTAV,  

5. FF,И,УР,Р1,ЕМ,Р2,ПДМ,НССС,ЕГ,СГ..READ,TAУ,И7,18,110,FF.,  

6. М1:=(DTAV**2+9*DTAV+18)/2,И1:=(18-ДТАУ-1)*(17-2),  

7. 'ВСТИ',INTEGER'ARRAY'ЧС/1:М1/,,,'ARRAY'ЧМК/1:FF,1:5/,,  

8. DDX,DX/1:17,1:18/),Х/1:М1:1:Н1/),У/1:М,1:Н1/),КК/1:М1/,,  

9. 'REAL'ОШ,ТЛ,ТХ,УВ,С1,С2,УМ,АИ,ОЛ,READЧ0,DX),,,  

10. FORM(Y,X,DTAV,DX,17,18,110,1),,'BEGIN'  

11. 'ARRAY'ЛХ/1:Н1/.,,'FOR'J:=1 STEP 1 UNTIL Н1 DO'AX(J,1):=,  

12. 'FOR'I:=1 STEP 1 UNTIL М1 DO'AX(I,J):=ABS(X(J,I)),,  

13. НОРМ(А,Х,Н1,С),КХ(J,1):=XS,,,'FOR'I:=1 STEP 1 UNTIL Н1 DO'  

14. Х(Ж,И):=Х(Ж,И)/XS'END,'END..,'FOR'I:=1 STEP 1 UNTIL Н1  

15. 'ДО'У(1,1):=Х(1,1)/НХ(1,1)..,БИ:='ЧС/0:М/,,,'FOR'I:=1 STEP 1  

16. 'ЧИЛ1:РМ'ДО'IF'И'Л'Т'И'О'К/1/,'THEN'БИ:=ЧС/1/,,,  

17. 'БЕСК',ARRAY'К01,К02,К03,К04/Ч1:3/,,CM/1:ЕМ/,HM1,HM2,  

18. ИМ3/1:3,1:4/,,СМАК/1:ИМ,1:2/,,ХХ,УУ/1:ЕМ,1:ИИ/,,  

19. ИМ2/1:ЕМ,1:3,1:РМ/,,СТ,СТРОС/0:М1/,,,'INTEGER'ARRAY'  

20. ИМ2/1:ИЛ,1:Н1,1:РМ/,,НСК/1:ЕМ/,,ИМ2/1:2*И-Н1/,,НМ1,  

21. ИМ2/1:ИЛ,1:Н1,1:РМ/,,НДК/1:И-ИЛ/,,НДК/1:М1/,,
```

```

22. 'PROCEDURE'HM(GM,XH,T),,'INTBXER'T.,'ARRAY'XH,IM.,'BCTR'-
23. , 'INTEGER'S,R.,S:=2.,R:=J.,B1M(/S,1/):=HM(/1,S/):=
24. XH(/R,1,T/),,HM(/S,S/):=XH(/R,2,T/),,IM(/S,4/):=XH(/R,3,T/),
25. S:=S+1.,R:=T., 'IF'SEQ'3'THEN'GOTO'B'END'.,
26. 'PROCEDURE'ON(NH,NO.), 'INTEGER'ARRAY,NH,NO., 'BEGIN'
27. , 'INTEGER'I,J,Z.,Z:=0..I:=1..NN.'FOR'J:=1'STEP'1'UNTIL'I'DO'
28. , 'IF'I'EQ(NH/J)'THEN'GOTOMM,Z:=Z+1..NO(/Z/):=I..MM:
29. I:=I+1., 'IF'I'LEM1'THEN'GOTO'NN'END'.,
30. YP:=1.,ECP2:=40(/0/),,P1:=40(/1/),, 'FOR'I:=1'STEP'1'UNTIL'BM
31. 'DO'FOR'J:=1'STEP'1'UNTIL'P'DO'NPK/I,1,J/):=0.,'PRINT'
32. , '(6HHOMEP YPABIEHHR,12)',YP., 'IF'P2EQ'P'THEN'BEGIN'
33. 'FOR'I:=1'STEP'1'UNTIL'P'DO'NDK/I/):=-P-I+1., 'GOTO'WEI'END'.,
34. 'BEGIN'ARRAY'D2/1:P/),,'FOR'J:=1'STEP'1'UNTIL'P'DO' BEGIN'
35. D2(J/):=0., 'FOR'I:=1'STEP'1'UNTIL'N1'DO'D2(J/):=D2(J/)*
36. X(J,I)*Y(YP,I/),,D2(J/):=ABS(D2(J/))'END'..PAH(M2,ND,P)
37. 'END'..WEI.'FOR'J:=1'STEP'1'UNTIL'P2'DO' 'FOR'I:=1'STEP'1
38. 'UNTIL'M1'DO'XX(J,I/):=X(AD(/P-J+1/),1/),,PR:=1.,IMM:=2..
39. PRH.'BEGIN'ARRAY'D2/1:N1/),,'INTEGER'ARRAY'ND/1:N1/),
40. 'FOR'I:=1'STEP'1'UNTIL'N1'DO' 'BEGIN'D2(J/):=0., 'FOR'J:=1
41. 'STEP'1'UNTIL'P2'DO'D2(J/):=D2(J/)*XX(J,I/)**2'END'.,PAH
42. (G2,ND,N1),, 'FOR'I:=1'STEP'1'UNTIL'I'DO' 'BEGIN'NMU(I/):=ND
43. (/1/),,NMU(I/):=NMU(NI-I+1/)'END'..ON(NM1,N01),,ON(NM2,N02)
44. 'END'..ET:=0..K:=0.. 'FOR'I:=1'STEP'1'UNTIL'I'DO' 'BEGIN'IT:-
45. NMU(I/),, 'FOR'J:=1'STEP'1'UNTIL'P'DO' 'IF'P'EQ'NMU2(J/)'THEN'
46. 'BEGIN'K:=P+1.,NMU(K/):=I., 'GOTO'EL'END'.,KJ:=END'., 'IF'PR
47. 'EQ'1'THEN' 'BEGIN'ARRAY'AY/1:N1/),,'FOR'I:=1'STEP'1'UNTIL'
48. N1'DO'AY(/I/):=ABS Y(YP,I/),,MAX(AY,N1,YM,1)END'., 'BEGIN'
49. 'ARRAY'XH/1:P2,1:3,1:3/),,'REAL'K21,H22,Y3,Y22,R:=H1-HL,Y3
50. :=SUM(R,1,1,N1,Y(YP,K/)),,Y22:=-SQR(SUM(R,1,1,N1,Y(YP,K)**2)
51. 2)/R1),, 'BEGIN'ARRAY'OB/1:R,1:2/),,'FOR'J:=1'STEP'1'UNTIL'R
52. 'DO' 'BEGIN'OB(J,I/):=H01(/J/),,OB(J,2/):=H02(/J/)'END'..
53. 'FOR'J:=1'STEP'1'UNTIL'P2'DO' 'BEGIN'XH(J,1,3/):=SUM(R,1,1,
54. N1,XH(/J,R/)),,XH(/J,2,3/):=SUM(R,1,1,N1,XX(/J,K)**2),,
55. XH(/J,3,3/):=SUM(R,1,1,N1,XX(/J,K/PY(YP,K/))), 'FOR'I:=1,2
56. 'DO' 'BEGIN'XH(/J,1,I/):=SUM(R,1,1,R,XX(/J,0B(R,I/)**2),,XH(J,3,L/):=
57. XH(J,2,L/):=SUM(R,1,1,R,XX(/J,0B(R,I/)**2),,XH(J,3,L/):=
58. SUM(K,1,1,R,XX(/J,0B(K,1,I/)**2)*Y(YP,0B(K,L,I/)**2))'END'END'
59. 'END'..XS:=XM:=0., 'FOR'I:=1'STEP'1'UNTIL'R'DO' 'BEGIN'XS:='IS-
60. Y(/YP,NO(/1/))..,XM:=XM+Y(/YP,NO2(/1/))'END'..W:=1..J:=0..
61. T,J:='J+1..1:=I+1..A, HM1(/1,1/):=HM2(/1,1/)..R.,HM1(/1,4/):=XS.,
62. HM2(/1,4/):=XM.,H21:=SUM(K,1,1,R,XX(/J,N01(/K/)))*XX(/1,N01
63. (/K/)),,H22:=SUM(K,1,1,R,XX(/J,N02(/K/)))*XX(/1,N02(/K/)))
64. ..HM1(/2,3/):=HM1(/3,2/):=H21..HM2(/2,2/):=HM2(/3,2/):=H22..
65. HM(/1,4/):=XM.,NM(HM1,XH,1)..NM(HM2,XH,2).. 'FOR'K:=1,2,3'DO'
66. 'FOR'S:=1,2,3,4'DO'HM3(/K,S/):=HM1(/K,S/)+HM2(/K,S/),
67. GAUSS(HM1,3,K01,H)..GAUSS(HM2,3,K02,H)..GAUSS(HM3,3,K03,H)..,
68. C1:=SUM(S,1,3,N1,(K01(/3/)-K02(/3/))*XX(/1,S/))**2)N1.,C2:=SUM(S,1,
69. XX(/J,S/)*(K01(/3/)-K02(/3/))*XX(/1,S/))**2)N1.,C2:=SUM(S,1,
70. 1,2*PI*H1,(YC/YP,NM1(S/))/K03(/1/)-K03(/2/)*XX(/J,NM1(S/))/-
71. K05(/3/)*XX(/1,NM1(S/))/)*2)/(2*PI*N1)..C:='SORT(C1+C2)/Y22.,
72. NM(HM3,XH,3)..HM(/1,1/):=H1..HM2(/1,4/):=Y3..HM3(/2,3/):=-
73. HM3(/3,2/):=SUM(HM1,1,1,N1,XX(/J,K)*XX(/1,K/))..GAUSS(HM3,3,
74. K04,H).. 'IF'P'EQ'TP1'THEN' GOTO'X..,QM(/W/):=C., 'FOR'R:=1,2,3
75. 'DO'KOG(/M,K,PR/):=KOG(/K/),,CMC(/M,1/):=C1,CMC(/M,2/):=C2.,
76. NK(/M,2,PR/):=I..NK(/M,3,PR/):=I.. 'IF'P'EQ'P1'THEN' PAH(CM,NC,
77. P1),,W:='W+1.. 'GOTO'W..M:='P'C., 'FOR'K:=1,2,3'DO'KOG(NK/P1/),,K,PR/):=
78. GG:=NC(/P1/),,CMC(/GG,1/):=C1/Y22**2..,CMC(/GG,2/):=C2/Y22**2.,
79. GM/NCK(P1/):=C.. 'FOR'K:=1,2,3'DO'KOG(NCK(P1/),,K,PR/):=
80. KOG(/K/),,NK(NCK(P1/),,2,PR/):=J..NK(NCK(P1/),,3,PR/):=I..PAH
81. (CM,NC,P1),, 'GOTO'W..M:='P'C., 'FOR'K:=1,2,3'DO'KOG(NCK(P1/),,K,PR/):=
82. 'GOTO'L., 'IF'J LT P-1'THEN' GOTO'G.. 'PRINT',(5X,6#ERROR,
83. 6#GAUSS=14)',ET., 'FOR'S:=1'STEP'1'UNTIL'P1'DO' 'BEGIN'OM:=0.,
84. 'FOR'J:=1'STEP'1'UNTIL'N1'DO' 'BEGIN'YB:=R09(/S,1,PR/),,FOR'
85. I:=2,3'DO'YB:=YB+KOG(/S,1,PR/)*XX(NCK(/S,1,PR/),,J/),,OM:=OM+
86. (Y(/YP,J/)-YB)**2'END'OM:='SORT(OMN1)/M.. 'IF'S EQ'1'THEN'
87. TX:=TJ:=OM.. 'IF'NC(1/)'EQ'S'THEN' 'BEGIN'AT:=OM.. 'IF'PR EQ'1
88. 'THEN' 'BEGIN'OL:=OM..PL:=I..NL:=S'END'.. 'IF'OL'GE'OM'THEN' Tl
89. 'BEGIN'OL:='OL..PL:=PL..NL:=S'END'.. 'END'.. 'IF'OW LT' TJ'THEN' Tl
90. ':=OM.. 'IF'OM ST' TX THEN' TX:=OM'END'.. 'PRINT'..(3X,4#RJ=,12,
91. 2,17#НВЫНО ОИЧАНИИ=12,,/5X,16#ИОКА3 ОМЕЛ МУШ=,E1 2,6,
92. 4X,5#ХУД=,E1 2,6,,/5X,17#ОЖЕА АЛПР ЛУЩ=,2E12,6,3X,5#ХУД=,
93. E12,6)',,PR,P1,CM/NC(/1/),,CM/NC(/P1/),,TJ,AL,TX,SH:NM

```

```

94. ::=2*IDM., 'FOR'J:=1'STEP'1'UNTIL'P1'DO,'FOR'I:=1'STEP'1'UNTIL'
95.   N1'DO' BEGIN'YY(J,I):=K03(J,1,PiV),., 'FOR'S:=2,3'DO'YY(J,
96.   I/):=YY(J,I/)*XX(NK(J,S,Pi/),I/)*K03(J,S,Pi/)'END'., 'FOR'
97.   J:=1'STEP'1'UNTIL'P1'DO,'FOR'I:=1'STEP'1'UNTIL'N'DO'XX(J,I/)'
98.   :=YY(J,I/), 'IF'PiEO'1'THEN' BEGIN'FOR'I:=1'STEP'1'UNTIL'
99.   FF'DO' BEGIN' IF'I'LE'P1'THEN' BEGIN'QML(I,1/):=M(NCK(I/))
100.   ..,CML(I,2/):=Pi.,CML(I,3/):=NCK(I/), 'FOR'J:=1,2'DO'
101.   CML(1,J+3/):=CML(NCK(1/),J/)'END'ELSE'CML(1,1/):=1000
102.   'END' END'ELSE' BEGIN'K:=1.,BE:I:=1.,C:=CML(1,1/),
103.   'FOR'J:=2'STEP'1'UNTIL'FF'DO,'IF'CML(1,J/):=C'R'C'THEN' BEGIN'
104.   C:=CML(J,1/),I:=J'END', 'IF'C'L'E'CM(NCK(K/))'THEN' GOTO,
105.   FIN.,CML(1,2/):=Pi.,CML(1,1/):=CM(NCK(K/)).., 'FOR'J:=1,2
106.   ,D0'CML(1,J+3/):=CM(NCK(K/),J/),..,CML(1,3/):=NCK(K/),..
107.   K:=K+1., 'IF'K'LE'P1'THEN' GOTO'BE, FIN.'END', 'IF'PiEO'DM
108.   'THEN' GOTO'J'END', Pi:=Pi+1.,P2:=P1..Pi:=40/Pi/), 'GOTO'
109.   PHi,J:'BEGIN', INTEGER'SS,, 'INTEGER', ARRAY'ST'(1:10,1:2/),
110.   'ARRAY'SK(1:10/),..,SS:=0.,HIM:=HIM-1.,HU:SS:=SS+1.,HP:Pi:=CML
111.   (/SS,2/),NCDG3:=CML(SS,3/), 'FOR'I:=1'STEP'1'UNTIL'10'DO'
112.   'BEGIN'ST'(1,I/):=ST(1,2/):=0.,SK(I/):=0'END', 'PRINT',(2X,
113.   6HМОДЕЛЬ,13,2X,5HРНЛ,12,3X,5HKOME,E11.5,3X,5HHEOM=E11.5,
114.   3X,4HКО=,E11.5,2H),..,SS,CML(1:SS,2/),CML(1:SS,1/),SORT(CML(1:
115.   4/),SORT(CML(1:SS,5/)),K:=0., 'FOR'I:=0'STEP'1'UNTIL'P DO',
116.   CTP0(I/):=CT(I/):=0., 'BEGIN', ARRAY'A,AD(1:PIDM/), 'INTEGER'
117.   ,ARRAY'AN,AND(1:IDM/),.., 'FOR'I:=1'STEP'1'UNTIL'IDM'DO,'BEGIN'
118.   A(I/):=AD(I/):=0..AN(I/):=AND(I/):=0'END', 'FOR'I:=1,2,3
119.   ,DO' BEGIN'A(I/):=K03(/NCDG3,I,Pi/),AN(I/):=NK(NCDG3,I,
120.   Pi/),END', 'IF'PiEO'1'THEN' GOTO'NN..,AD(1/):=A(1/),..,MM:
121.   MM:=1., 'FOR'I:=2'STEP'1'UNTIL'IDM'DO,'IF'A(I/)'NE'0'THEN'
122.   'BEGIN', 'IF'AN(I/)'EQ'0'THEN' BEGIN'W:=W+1.,AD(W/):=A(I/),
123.   AND(W/):=0'END'ELSE' BEGIN' FOR'S:=1,2,3'DO' BEGIN'AD(W+
124.   S/):=A(I/)*K03(AN(I/),S,Pi-1/),..,AND(W+S/):=NK(AN(I/),S,
125.   Pi-1/)'END',..,W:=W+3'END'END', 'FOR'I:=2'STEP'1'UNTIL'IDM'DO'
126.   'BEGIN'A(I/):=AD(I/),..,AN(I/):=AND(I/)'END',..,Pi:=Pi-1.,
127.   'IF'PiGT'1'THEN' GOTO'MA,NN.'FOR'I:=1'STEP'1'UNTIL'IDM'DO'
128.   'BEGIN'S:=AN(I/),CT(S/):=CT(S/)*A(I/)'END', 'FOR'I:=1
129.   'STEP'1'UNTIL P'DO'CTPO(AD(1:1/)):=CT(1/)'END',..,CTP0

```

130. U(0/):=CT(0/)\*KK(YP/),..,R:=S:=0., 'PRINT' '(1ЭНКСОФФИЦИЕНТ',
131. /,2KEM2.5)', CTP0(0/),.., 'FOR'J:=1'STEP'1'UNTIL'M1+PN'DO', 'FOR'
132. I:=0'STEP'1'UNTIL'TAY(J/)'DO,'BEGIN'S:=S+1., 'IF'CTPO(S/)'NE'
133. 0'THEN' BEGIN'CTPO(S/):=CTPO(S/)'END', 'IF'K'EQ'10
134. ST(K,1/):=J.,ST(K,2/):=I.,SK(K/):=CTPO(S/),.., 'IF'K'EQ'10
135. 'THEN' BEGIN'PRINT' '(10(X,XHMA=,I3))', 'FOR'K:=1'STEP'1
136. 'UNTIL'10'DO'ST(K,1/), 'PRINT' '(10E2.5)', SK., 'FOR'K:=1
137. 'STEP'1'UNTIL'10'DO,'BEGIN'ST(K,1/):=ST(K,2/):=SK(K/);=0
138. 'END',..,K:=0'END', 'END',.., 'IF'K'NE'0'THEN' BEGIN' PRINT'
139. '(10(6X,XHMA=,I3))', 'FOR'K:=1'STEP'1'UNTIL'10'DO'ST(K,1/),..,
140. 'PRINT' '(10E12.5)', SK'END',..,C:=0.,J:=0., 'FOR'I:=1'STEP'1
141. 'UNTIL'IT'DO,'FOR'J:=1'STEP'1'UNTIL'IS'DO'DDX(I,J/):=
142. DDX(1,J/), 'FOR'I:='DIAV+2'STEP'1'UNTIL'1B'DO,'BEGIN'
143. 'FOR'K:=2'STEP'1'UNTIL'IT-1'DO,'BEGIN'FORM(Y,X,
144. DTAV,DDX,IT,IB,110,0),..,J:=(K-2)\*(18-DTAV-1)+I-DTAV-1.,CI:-
145. CTP0(K/),.., 'FOR'S:=-1'STEP'1'UNTIL'P'DO'CI:=CI+CTPO(S/)+
146. X(S,J/),.., 'IF'ABS(CI)'GT'1000'THEN' BEGIN'C:=10000., 'PRINT'
147. '(5X,20HМОДЕЛЬ НЕСТОЧИЯ)',.., 'GOTO'JN'END',..,DDX(K,I/):=
148. C1'END', 'END',..,C:=0., 'FOR'I:=2'STEP'1'UNTIL'IT-1'DO,'FOR'J:-
149. DTAV+2'STEP'1'UNTIL'1B-1'DO'C:=C+(DDX(1,J/)-DX(I,J/))\*2.,
150. 'PRINT' '(2X,12HКОЕ.+ИНГ.E14.5./,2X,6HИНГ.=E14.5.),..,
151. SORT(C/M1+CML(1:SS,1/)),SORT(C/M1)..,
152. 'PRINT' '(5X,25HРЕЗУЛЬТАТ ИНТЕРПОЛАЦИЯ)',..,
153. 'FOR'I:=1'STEP'1'UNTIL'IT'DO' 'PRINT' '(4X,15FB.4)',..,
154. 'FOR'J:=-1'STEP'1'UNTIL'18-1'DO'DDX(I,J/),..,
155. JN:'PRINT' '(50(H-),/)',.., 'IF'SSL'FF'THEN' GOTO'HH..,
156. YP:=YP+1., 'IF'YP'LE'W'THEN' GOTO'EO'END', 'END', 'END', 'EOF',
157. 'PROCEDURE FORM(Y,X,TAY,DX,11,12,13,PP),..,
158. 'ARRAY'Y,X,DX,, 'INTEGER',TAY,11,12,13,PP., 'BEGIN', 'INTEGER'
159. I,J,L,I1,J1,J2,J3,J4,L2,K,, 'INTEGER',ARRAY'T/(1:3,1:10),..,
160. K:=0., 'FOR'I:=2'STEP'1'UNTIL'11-1'DO,'BEGIN', 'IF'IE0'13'THEN'
161. 'GOTO'P11., 'FOR'J:=1'STEP'1'UNTIL'12-TAY-1'DO,'BEGIN'
162. J1:=J+K,Y/(I,J/):=DX(I,J-TAY+1/),.., 'FOR'L:=1'STEP'1'UNTIL'
163. TAY+1'DO' BEGIN'Y(I,J/):=DX(I,J-TAY+1-L/)'END',..,J2:=1.,
164. 'FOR'L:=1,2'DO' BEGIN'J2:=-J2..X(TAY+1-L,J1/):=DX(I+J2,
165. J+TAY/)'END',..,J3:=J2+TAY+3., 'FOR'L:=1'STEP'1'UNTIL'J2'DO'

```

166. 'FOR' I4:=L'STEP'1'UNTIL'J2'DO" BEGIN' J3:=J3+1., X(J3,J4):=
167. X(L,J1)*X(L,J4)'END' END', K:=K+12*TAY-1..P1:'END'.,
168. 'IF' P'R'E0'0'THEN' GOTO'015., PRINT'"/12НРАЧУМРОВКА ,
169. 10ННЕРЕЧЕНЫХ"/)0.., J4:=1.., FOR'I..I-1'STEP'1'UNTIL'J2'DO"
170. 'BEGIN' J4:=J4.., PRINT'/(2X,12, Ji - ), I., IF I'LE'TAY+1
171. 'THEN' PRINT'/(1H..TX,ЭНХ(0).., 'IP'1'СР'TAY+1'THEN' PRINT'
172. ''(1H..TX,2H(X,12,ЭНХ,0))', J4., IF I'LE'TAY+1
173. ''(1H+,10X,1H,,12,1H))', 1-I'END', K:=0..L1:=J2.., FOR'I:=1
174. 'STEP'1'UNTIL'J2'DO"FOR'L:=1'STEP'1'UNTIL'J2'DO" BEGIN' K:=
175. K+1.., L1:=L1+1.., T(1,H):=L1..T(2,K):=1..T(3,K):=L1.., IP'K
176. 'EQ'10'THEN' BEGIN' K:=0.., PRINT'/(2X,10(13,2H-X,12,2H X,12,
177. X)), 'T'END' 'END'., 'IF' P'NE0'THEN' PRINT'/(2X,10(13,2H-X,12,
178. 2H X,X))', T.., FOR'I:=1'STEP'1'UNTIL'J2'DO" PRINT'/(4X,
179. 10F10.4)', 'FOR'J:=1'STEP'1'UNTIL'(12-TAY-1)*(11-2)'DO"
180. X(1,J).., PRINT'/(4X,10F10.4)', Y.., M5:'END' 'EOP'
181. 'ROCEDURE' NORM(XN,N,P), 'ARRAY' XN.., 'INTEGER' N.., 'REAL' P.,
182. 'BEGIN' 'INTEGER' K., 'REAL' XA.., MAX(XN,N,XM,K).., P:=1..
183. NN:=P*10.., IF P>T(XM)'THEN' GOTO'PP.., GOTO'NN.., PP:P:=P/10.,
184. 'IF' P>T(XM)'THEN' GOTO'M.., GOTO'PP.., IF P=P*10'END' 'EOP',
185. 'PROCEDURE' РАН(Х,NP,N), 'ARRAY' X.., 'INTEGER' ARRAY' NP.,
186. 'INTEGER' N.., 'BEGIN' 'ARRAY' XD/(1:N)/.,
187. 'INTEGER' ARRAY' ND/(1:N)/.., REAL XM.., 'INTEGER' K, IN1.,
188. 'FOR' I:=1'STEP'1'UNTIL'N'DO" BEGIN' XD(I/1):=X(I/1)..,
189. ND(I/1):=I'END'.., N1:=N.., MM:=MAX(XD,N1,XM,K)..,
190. NP(NL):=ND(X/K).., 'FOR' I:=K+1'STEP'1'UNTIL'N1'DO"
191. 'BEGIN' XD(I-1/1):=XD(I/1).., ND(I-1/1):=ND(I/1)'END'.., N1:=N1-1.,
192. 'IF' N1'GE'2'THEN' GOTO'M.., NP(V/1):=ND(V/1)'END' 'EOP'
193. 'PROCEDURE' GAUSS(A,N,X,ERROR)..,
194. 'VALUE' N.., 'ARRAY' A,X.., 'INTEGER' N.., 'LABEL' ERROR.,
195. 'BEGIN' 'REALT.., 'INTEGER' I,J,K,L,M,L:=0.., CO:L:=L+1.,
196. 'FOR' K=L'STEP'1'UNTIL'N'DO" IF ABS(A(/K,L))>T0.00000001
197. 'THEN' GOTO'G1.., GOTO'ERROR.., G1:'IF' K=L'THEN' GOTO'C2.,
198. 'FOR' M:=L'STEP'1'UNTIL'N+1'DO" BEGIN' T:=A(/L,M)..,
199. A(/L,M):=A(/K,M).., A(/K,M):=T'END'.,
200. C2:'FOR' J:=N+1'STEP'-1'UNTIL'L'DO"
201. A(/L,J):=A(/L,J)/A(/L,L).., 'FOR' I:=K+1'STEP'1'UNTIL'N'DO"

```

```

202. 'BEGIN' T:=A(/I,L).., 'FOR' J:=I+1'STEP'1'UNTIL'N+1'DO"
203. A(/I,J):=A(/I,J)-A(/L,J)*T'END'., 'IF' L'NE'N'THEN'
204. 'GOTO' C0.., 'FOR' I:=N'STEP'-1'UNTIL'1'DO"
205. 'BEGIN' X(/I):=A(/I,N+1).., 'FOR' K:=I-1'STEP'-1'UNTIL'1'DO"
206. A(/K,N+1):=A(/K,N+1)-A(/K,I)*X(/I)'END' END' 'EOP'

```

### 3. ПРОГРАММА ИДЕНТИФИКАЦИИ ДИНАМИКИ ПОЛЕЙ СЛОЖНЫХ ОБЪЕКТОВ НА ОСНОВЕ КОНЕЧНО-РАЗНОСТНОЙ АППРОКСИМАЦИИ

Программа (прогр. 48) предназначена для идентификации трехмерной области (полигонов) по данным натурных наблюдений [32].

Язык программирования АЛГОЛ, ЭВМ БЭСМ-6.

**Используемые функции и подпрограммы.** COMBI (см. гл. 3)

**Метод.** В основу идентификации динамики положены принципы эвристической самоорганизации и уравнение баланса масс поля минимизации:

$$\frac{\partial M}{\partial t} + \sum_{\alpha=1}^3 V_\alpha \frac{\partial M}{\partial X_\alpha} = \sum_{\alpha=1}^3 \frac{\partial}{\partial X_\alpha} \left( K_\alpha \frac{\partial M}{\partial X_\alpha} \right) + P(\bar{x}), \quad (4.7)$$

где  $\bar{x} = (x_1, \alpha = 1, 2, 3); P(x)$  — полином не выше второй степени (остаток) вида

P(\bar{x}) = a\_0 + a\_1 x\_1 + a\_2 x\_2 + a\_3 x\_3 + a\_4 x\_1^2 + a\_5 x\_2^2 +

$$+ a_6 x_3^2 + a_7 x_1 x_2 + a_8 x_1 x_3 + a_9 x_2 x_3. \quad (4.8)$$

Программа автоматически составляет необходимые разности по шаблону (4.9). Линейное дифференциальное уравнение в частных производных (4.7) аппроксимируется алгебраическим конечно-разностным уравнением из требования первого порядка аппроксимации по  $t$  (времени) и второго порядка по  $h_\alpha$ . Таким образом, точность аппроксимации полученного конечно-разностного уравнения имеет порядок 0 ( $\tau, h_1^2 + h_2^2 + h_3^2$ ), где  $h_1, h_2, h_3$  — шаги по соответствующим координатам. Аппроксимация линейного дифференциального уравнения в частных производных (4.7) соответствует конечно-разностным уравнением на равномерной сетке имеет вид

\frac{M\_{i,j,k}^{t+1} - M\_{i,j,k}^t}{\tau} + V\_k \left( \frac{M\_{i-1,j,k}^t - M\_{i,j-1,k}^t}{2h} \right) +

$$+ V_y \left( \frac{M_{i,j+1,k}^t - M_{i,j-1,k}^t}{2h} \right) + V_z \left( \frac{M_{i,j,k+1}^t - M_{i,j,k-1}^t}{2h} \right) = \\ = K_x \left( \frac{M_{i+1,j,k}^t + M_{i-1,j,k}^t + 2M_{i,j,k}^t}{h^2} \right) + K_y \left( \frac{M_{i,j+1,k}^t + M_{i,j-1,k}^t + 2M_{i,j,k}^t}{h^2} \right) + \\ + K_z \left( \frac{M_{i,j,k+1}^t + M_{i,j,k-1}^t - 2M_{i,j,k}^t}{h^2} \right) + P(x); \quad (4.9)$$

$$t = n\tau; \quad x_1 = ih; \quad y_1 = jh; \quad z_1 = kh; \quad x_2 = y; \quad x_3 = z,$$

где  $V_x, V_y, V_z$  — компоненты скорости;  $K_x, K_y, K_z$  — коэффициенты диффузии;  $P(x)$  — остаток, выраженный в виде функции координат.

Определяется оптимальная структура остатка. Например, если  $P(\bar{x})$  представляется собой полином первой степени, то уравнение (4.9) можно привести к следующему виду:

$$a_1 + V_x a_2 + V_y a_3 + V_z a_4 = K_x a_6 + K_y a_7 + b_8 x + b_9 y + b_{10} z, \quad (4.10)$$



#### 4. ПРОГРАММА ИДЕНТИФИКАЦИИ МНОГОМЕРНЫХ ПОЛЕЙ ПО НЕРЕГУЛЯРНОЙ СЕТКЕ

Предназначена для идентификации динамических систем с распределенными параметрами по наблюдениям, заданным на регулярной по времени и пространству сетке, при помощи модели в виде шаблона с запаздываниями, линейной по ходящим в нее коэффициентам (прогр. 49)

Язык программирования АЛГОЛ, ЭВМ БЭСМ-б

**Используемые подпрограммы.** COMB\_I

**Метод.** Записем уравнение поля в конечно-разностной форме:

$$\frac{U_{i,j}^{k+1} - U_{i,j}^k}{t_{j+1} - t_j} = a_n + \sum_{n=1}^N a_n U_{i,n}^k + B \sum_{n=1}^M \frac{U_{i,j}^k - U_{i,q}^k}{\varphi(\rho_{q,j})} + \sum_{r=1}^{ZA} C_r U_{i,r}^{k-1}. \quad (4.11)$$

где  $U_{i,j}^k$  —  $i$ -й параметр ( $i$ -я компонента) многомерного поля в  $k$ -й момент времени в  $j$ -й точке пространства,  $t_k$  —  $k$ -я точка на оси времени;  $N$  — количество параметров, характеризующих поле,  $M$  — количество точек в пространстве (их координаты предполагаются постоянными во времени),  $ZA$  — количество запаздываний;  $\rho_{q,j}$  — евклидово расстояние между  $q$ -й и  $j$ -й точками;  $\varphi$  — лекоторая функция от расстояния. В программе принято

$$\varphi(\rho) = \begin{cases} \rho, & \text{если } \rho \neq 0; \\ 1, & \text{если } \rho = 0. \end{cases}$$

Из уравнения (4.11) видно, что в правую часть каждого из  $N$  уравнений системы входит не более, чем  $1 + N + 1 + ZA$  членов. Члены со второго по  $(N + 1)$ -й с коэффициентами  $a_n$  — значения каждого из  $N$  параметров поля в данной точке в данный момент времени,  $(N + 2)$ -й член с коэффициентом  $B$  является некоторой конечно-разностной аппроксимацией первой производной по пространству. При отсутствии регуляризации невозможно вычислить конечно-разностную аппроксимацию первой и второй производных по каждому из  $R$  направлений пространства, поэтому произведена их замена единственным членом,

$$\sim \sim \sim \sum_{q=1}^M \frac{U_{i,j}^k - U_{i,q}^k}{\varphi(\rho_{q,j})}, \quad (4.12)$$

который при  $R = 1$  и  $\varphi(\rho) = \rho$  выражается в усредненную по всем точкам аппроксимацию первой производной по пространству в данной точке. Члены с  $(N + 1) + ZA + 2$ -й с коэффициентами  $C_r$ , представляют собой значения данного параметра в данной точке пространства, взятые с запаздываниями по времени от одного такта до  $Z$  тактов. Наличие этих членов физически может соответствовать увеличению порядка производной по времени соответствующего уравнения системы

Таким образом, уравнение (4.11) позволяет, минуя этап интерполяции регуляризации сетки по нерегулярным данным, непосредственно идентифицировать динамическую систему с распределенными параметрами. При этом учитываются все члены уравнения

**Описание параметров.** *Вводимое:*

$N$  — количество параметров поля (размерность поля);

$TM$  — количество точек в пространстве;

$ZA$  — количество запаздываний в уравнении (4.11);

устанавливаемое пользователем;

$NU$  — число независимых параметров;

$R$  — размерность пространства (но не размерность поля);

$CB$  — количество лучших по критерию регуляризации выводимых на печать уравнений по каждому из  $N$  параметров;

АТ [1 : TM] — массив точек на оси времени;  
 У [1 : TM, 1 : M, 1 : N] — массив координат точек в пространстве;  
 ТУ [1 : M \* (TM - ZA - 1)], PS [1 : 2 + N + ZA, 1 : CB] и OS [1 : CB] совпадают с однотактными, описанными в подпрограмме COMBI (прогр. 31) (3.4), FO [1 : M] — массив производов по каждой из  $M$  точек пространства. Вычисляется  $N \times CB$  раз — по каждой из  $CB$  моделей и по каждому из  $N$  параметров поля; МП, МЭ — количество точек проверочной и экзаменационной последовательностей соответственно.

**Пользовательность работы программы.** Последовательно заполняются столбцы матрицы  $X$  (строки 22—32). Массив  $U$  заполняется значениями конечно-разностной аппроксимации первой производной по времени данного параметра поля в данной точке (строки 33—35).

Обращение к прогрессии COMBI, в результате применения которой находятся СВ лучших уравнений для данного параметра (строки 36—37). Для получения системы уравнений необходимые операции алгоритма выполняются в цикле  $N$  раз по числу параметров поля.

**Примечания:** 1.  $N + ZA \leqslant 16$ .

2. Особенностью программы является ее целенаправленность на идентификацию полей по нерегулярным сеткам исходных данных. Регулярная сетка не является частным случаем по отношению к описанному случаю нерегулярной сетки.

3. В приведенной программе между обучающей проверочной и экзаменационной последовательностями состоянию равно 6 : 3 : 1. При необходимости оно может быть изменено путем изменения строки 7.

4. В программе предусмотрено  $\varphi(\rho) = \rho$ , т. е. влияние точки поля обратно пропорционально расстоянию до нее. Если же из физических соображений следует, что поле имеет другой закон распределения  $\varphi(\rho)$ , то необходимо соответственно изменить строку 28.

5. Все изменения, допустимые в подпрограмме COMBI, не нарушают работу программы.

#### ПРОГРАММА 49.

```

1.   'ALGOL';'COMBI..'
2.   'BEG IN', INTEGER,N,M,TM,ZA,R,T,I,J,K,G,';',CB,PRY,NU..
3.   'REAL',TU,S3.,READ(N,M,TM,ZA,R,CB),,READ(TU),,
4.   ',BEGIN','ABRA',U/1:TM,1:M,1:N/,AT(1:TM),AC(1:R,1:M/),
5.   PS(1:(2+N+ZA):1:CB/),OS(1:1:CB/),FO(1:M/),
6.   X(1:M*(TM-ZA-1),1:(1:N+ZA)),Y(1:M*(TM-ZA-1))/,PRY:=0.,
7.   NU:=0.,MT:=ENTER.'#M*(TM-ZA-1)*MT',
8.   READ(AT,AC,U),,PRINT'',(5X,1N)КОЛИЧЕСТВО ПАРАМЕТРОВ,13)'',N,
9.   'PRINT'',(5X,1N)КОЛИЧЕСТВО ПРОСТРАНСТВА,13)',M,
10.  'PRINT'',(5X,1N)КОЛИЧЕСТВО СЛОВ,13)',TM,
11.  'PRINT'',(5X,1N)КОЛИЧЕСТВО ЗАПАЗДЫВАНИЙ,13)',ZA,
12.  'PRINT'',(5X,24)РАЗМЕРНОСТЬ ПРОСТРАНСТВА,13)',R.,
13.  'PRINT'',(5X,14)СВОБОДА ВЫБРА,13)',CB,
14.  'PRINT'',(5X,14)СВОБОДА ВЫБРА,13)',CB,'

```

15. 'UNTIL' TM'DO'AT( T / ) .. , 'PRINT' .. '(/5X5ПСЕТА)' .. , 'FOR' J:=1

16. 'STEP' 1' UNTIL' R'DO' , BEG IN' 'PRINT' .. '(/5X,11НКООРДИНАТ, T2)' .. , J,,

17. 'PRINT' .. '(12E10.4)' .. , 'FOR' I:=1'STEP' 1' UNTIL' M'DO'

18. AC( /J, I / ) END' .. , 'PRINT' .. '(/5X,16ИСХОДНЫЕ ДАННЫЕ ,/A, 17(1H-))'

19. .. , 'FOR' J:=1'STEP' 1' UNTIL' N'DO' , BEG IN' 'PRINT' .. '(/5X, 9НДАРЕИР,

20. I2)' .. , J,, 'FOR' I:=1'STEP' 1' UNTIL' M'DO' , 'PRINT' .. '(12E10.4)',

21. 'FOR' T:=1'STEP' 1' UNTIL' TM'DO' , BEG IN' 'PRINT' .. '(/5X,

22. 20ИЧСЛО АРГУМЕНТОВ L=, I3)' .. , 2+N+ZA,, 'FOR' K:=1'STEP' 1' UNTIL' N

'DO' , 'FOR' T:=1+ZA'STEP' 1' UNTIL' TM-1'DO' , 'FOR' I:=1'STEP' 1

'UNTIL' M'DO' X( /I+(T-ZA-1)\*M, K / ) :=U( /T, I, K / ) .. , 'FOR' G:=1'STEP' 1

'UNTIL' N-HU'DO' , BEGIN' , 'FOR' T:=1+ZA'STEP' 1' UNTIL' TM-1'DO'

'FOR' J:=1'STEP' 1' UNTIL' M'DO' , BEGIN' , 'REAL' S1, S2, P0, OM, SI:=0,,

, 'FOR' I:=1'STEP' 1' UNTIL' M'DO' , BEG IN' S2:=0,, , 'FOR' K:=1'STEP' 1

28. 'UNTIL' R'DO' S2:=S2+(AC( /K, I / )-AC( /K, J / ))\*2,, , OM:=P0:=SORT(S2),,

29. 'IF' OM=0'THEN' OM:=1, SI:=S1+(U( /T, I, G / )-U( /T, J, G / ))/OM

30. , 'END' .. , X( /J+(T-ZA-1)\*M, N+1 / ) :=SI'END' .. , 'FOR' K:=1'STEP' 1

'UNTIL' ZA'DO' , 'FOR' T:=1+ZA'STEP' 1' UNTIL' TM-1'DO' , 'FOR' I:=1

'STEP' 1' UNTIL' M'DO' X( /I+(T-ZA-1)\*M, N+K+1 / ) :=U( /T-K, I, G / ) ..

33. 'FOR' T:=1+ZA'STEP' 1' UNTIL' TM-1'DO' , 'FOR' I:=1'STEP' 1' UNTIL' M

'DO' Y( /I+(T-ZA-1)\*M / ) :=(U( /T+1, I, G / )-U( /T, I, G / ))/(AT( /T+1/)

-AT( /T / )) .. , 'PRINT' .. '(/,7X,23ИДЕТИФИКАЦИЯ УРАВНЕНИЯ, I3/7X,

36. 23(4H\*)' .. , G,, , COMBI( 1, 1+N+ZA, M\*( TM-ZA-1 ), M1, M2, CSV, 0, 1, PRY,

37. X, Y, PS, OS ), , 'PRINT' .. '(/13X,14ЧИЩУЩИЕ МОДЕЛИ:/12X,16(1H-))' .. ,

38. 'FOR' I:=1'STEP' 1' UNTIL' CB'DO' , 'PRINT' .. '(/7X,6ИМОДЕЛЬ, Г3, 5X,  
39. 17ИЗНАЧЕНИЕ КРИТЕРИИ, E15.5/(10E12.5))' .. , I, OS( /V / ), 'FOR' J:=1  
40. , 'STEP' 1' UNTIL' N+ZA-2'DO' PS( /J, I / ) .. , 'IF' TU'NEQ'0'THEN' , 'BEGIN'  
41. 'PRINT' .. '(5X,24НДР ОПНОЗ С УПРЕДЕЛЕНИЕМ ИА, E10.2)' .. , TU,, 'FOR' J:=1  
42. , 'STEP' 1' UNTIL' CB'DO' , BEG IN' S3:=0,, , 'FOR' K:=1'STEP' 1

43. , 'UNTIL' 1+N+ZA'DO' S3:=S3+X( /M\*( TM-ZA-1 ), K / )\*PS( /K+1, J / ) ..

44. S3:=S3+PS( /I, J / ) .. , 'FOR' I:=1'STEP' 1' UNTIL' M'DO' , 'FOK' /I / ) :=

45. U( /TM, I, G / )+TU\*S3 .. , 'PRINT' .. '(5X,17НДР ОПНОЗ ПО МОДЕЛИ, I3)' .. , J,,

'PRINT' .. '(12E10.4)' .. , 'FOR' I:=1'STEP' 1' UNTIL' M'DO' , 'FOK' /I / )

46. , 'PRINT' .. '(12E10.4)' .. , 'FOR' I:=1'STEP' 1' UNTIL' M'DO' , 'FOK' /I / )

47. , 'END' .. END' , END' , END' , EOF

## Глава 5

### СИНТЕЗ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

### И ИССЛЕДОВАНИЕ УСТОЙЧИВОСТИ

#### 1. ЗАДАЧА ОПТИМАЛЬНОГО УПРАВЛЕНИЯ И ИССЛЕДОВАНИЯ УСТОЙЧИВОСТИ

Исследуется устойчивость линейных дискретных систем с запаздываниями и синтезируется оптимальное управление на скользящем интервале как для линейных, так и для нелинейных систем, содержащих наряду с динамическими ограничениями, накладываемые на переменные состояния и управления в форме равенств и неравенств.

Рассматриваются дискретные системы вида

$$q_{k+1} = f(q_k, q_{k-1}, \dots, q_{k-\tau_1}, u_k, u_{k-1}, \dots, u_{k-\tau_2}, \lambda_k), \quad (5.1)$$

где  $q_i^T = (q_i^1, q_i^2, \dots, q_i^n)$  —  $n$ -мерный вектор состояний;  $u_i^T = (u_i^1, u_i^2, \dots, u_i^m)$  —  $m$ -мерный вектор управлений;  $\lambda_i^T = (\lambda_i^1, \lambda_i^2, \dots, \lambda_i^l)$  —  $l$ -мерный вектор возмущений;

$f^T = (f^1, f^2, \dots, f^n)$  —  $n$ -мерная вектор-функция; нижний индекс означает соответствующий момент времени.

Программ. 50 предназначена для исследования устойчивости системы (5.1), программы 51—52 — для синтеза оптимального управления системой вида (5.1) при дополнительных ограничениях в форме неравенств вида

$$g(q) \geqslant 0; \quad (5.2)$$

$$G(u) \geqslant 0 \quad (5.3)$$

( $g$ ,  $G$  — соответственно  $n_1$ - и  $m_1$ -мерные выпуклые вектор-функции; нули справа — векторы соответствующих размерностей), а также ограничения в форме равенств вида

$$h(q) = 0; \quad (5.4)$$

$$H(u) = 0 \quad (5.5)$$

( $h$ ,  $H$  — соответственно  $n_2$ - и  $m_2$ -мерные вектор-функции; нули — векторы соответствующих размерностей).

Задача синтеза оптимального управления для системы вида (5.1) состоит в том, что для функционала качества

$$\Phi = \sum_{k=K_0}^{K_0+L} \Psi(q_k, u_k) \quad (5.6)$$

ищется такая последовательность управлений  $u_{k_0}^*, u_{k_0+1}^*, \dots, u_{k_0+L}^*$ , что при линейном ограничении (5.1), а также при условии выполнения ограничений (5.2) — (5.5) значение функционала (5.6) достигает своей нижней границы. Здесь  $L$  — интервал оптимизации, который может быть фиксированным (стандартная задача) или скользящим (при переменном  $K_0$ ). Управление с оптимизацией на скользящем интервале состоит в том, что на каждом шаге движения системы решается задача управления (5.6), но на объекте реализуется лишь одно, первое значение управления  $u_{k_0}^*$ , а для выбора следующего значения  $u_{k_0+1}^*$  задача синтеза оптимального управления решается снова, причем минимизирующим функциона.

лом теперь является

$$\Phi = \sum_{k=K_0+1}^{K_0+L+1} \varphi(q_k, u_k). \quad (5.7)$$

Аналогичная процедура повторяется на всем интервале управления. Такая постановка задачи является достаточноющей.

Управление на скользящем интервале имеет ряд преимуществ [34]. Основные из них: уменьшаются вычислительные затраты, так как обычно без ущерба для оптимальности интервал  $\bar{I}$  можно принять существенно меньшим всего интервала управления (который может быть бесконечным); Можно оперативно на каждом шаге учитывать изменение прогноза возмущений; исследуемая система (5.1) более устойчива к неконструируемым возмущениям и неточностям математического описания объекта. Последнее означает, что для нелинейной или нестационарной объекта можно оптимизировать, применив стационарную аддитивную модель более низкого порядка, аппроксимирующую динамику объекта на ограниченном интервале.

Краткая характеристика программ, приведенных в настоящей главе Прогр. 50 предназначена для идентификации линейного конечно-разностного описания линейной системы с запаздываниями и для исследования ее устойчивости. Идентификация осуществляется на основе многоядного алгоритма МГУА с линейными частными описаниями, а исследование устойчивости получаемой линейной системы основано на анализе абсолютных величин корней характеристического многочлена. Возможна параллельное исследование устойчивости различных подсистем, состоящих из полных систем.

Прогр. 51—52 решают задачу синтеза оптимального управления. Все они ориентированы на оптимизацию на скользящем интервале, но применимы и для случая фиксированного интервала. В этих программах задача оптимизации решается следствием ее к эквивалентной задаче нелинейного программирования. Реализованный алгоритм оптимизации является модификацией алгоритмов проектирования градиента с восстановлением связей. Обе эти программы допускают введение дополнительных ограничений на переменные состояния и управления. В прогр. 51 могут быть учтены ограничения вида  $a \leq q \leq b$  и  $c \leq u \leq d$ , что является частным случаем ограничений (5.2) — (5.3).

Прогр. 52 предназначена для синтеза оптимального управления линейными динамическими системами. Эта программа, являясь специализированной, может быть применена (при соответствующих изменениях) для синтеза оптимального управления нелинейными системами и допускает любые ограничения (5.2), (5.3).

## 2. ПРОГРАММА ИДЕНТИФИКАЦИИ ЛИНЕАРИЗОВАННЫХ УРАВНЕНИЙ ДИНАМИКИ СЛОЖНЫХ СИСТЕМ И ИССЛЕДОВАНИЯ ИХ УСТОЙЧИВОСТИ

Предназначена для объективной идентификации сложных систем методами самоорганизации и для исследования их устойчивости [9] (прогр. 50). Язык программирования АЛГОЛ, ЭВМ БЭСМ-б

**Используемые подпрограммы.** MAX, NORM, KOCH, GAUSS, РАНЖ.

**Метод.** Восстановление характеристического уравнения для расширенной системы производится по методу Фаддеева, а вычисление корней характеристического уравнения — по методу Воеводина.

Процесс построения аппроксимирующей функции (полного описания) является обычной реализацией многоядного алгоритма МГУА с линейными частными описаниями. В качестве критерия селекции используется минимум смещения по коэффициентам, определяемый формулой (2.9).

**В** первую обучающую последовательность для отыскания коэффициентов  $a_i$  выделяются точки, имеющие наибольшее значение дисперсии по множеству первичных. Вторая обучающая последовательность для вычисления коэффициентов  $\bar{b}_i$  содержит все точки, не вошедшие в первую последовательность, а также точки с наименьшей дисперсией, попавшие в нее.

### Описание параметров:

**N** — длина исходной последовательности данных;  
**M** — число зависимых переменных, образующих систему;  
**M1** — число управляющих воздействий;  
**MII** — размерность первой подсистемы;  
**P** — длина проверкой последовательности;  
**RM** — максимальное число рядов селекции;

**XI[1 : M+M1, 1 : N]** — массив значений  $M + M1$  переменной в  $N$  точках;  
**TAУ[1 : M+M1]** — вектор значений максимально учитываемых запаздывающих;

**ЧО [0 : RM]** — вектор, задающий изменение свободы выбора по рядам селекции.

**Последовательность работы программы.** После ввода исходной информации значения зависимых и независимых переменных центрируются (при необходимости), нормируются и подготавливаются к использованию в многоядной процедуре МГУА (осуществляется предварительный отбор наиболее важных переменных, формируются необходимые информационные массивы). Далее для каждой из зависимых переменных, в соответствии с процедурой МГУА, строится многоядная линейная структура частных моделей, а уравнения оптимальной стоянки представляются в развернутом виде. Таким образом формируется система линейных разностных уравнений, описывающих динамику идентифицируемого объекта. После этого исследуется устойчивость полной системы и подсистем заданной размерности, образующих полную систему в случае их взаимосвязи.

**Примечания:** 1. Система линеаризованных уравнений динамики строится в разностной форме, каждое уравнение выражает будущее значение переменной (или ее приращение) через линейную комбинацию прошлых значений всех аргументов со всеми учитывающими запаздываниями.

2. Оптимизация сложности модели (число рядов селекции) производится отдельно для каждого уравнения системы.

3. Выбор модели осуществляется по результатам просчета заданного числа рядов.

4. Слобода выбора может захватываться от ряда к ряду произвольно.

5. Обучающие последовательности из исходной выборки назначаются заново для каждого уравнения и ряда селекции.

### ПРОГРАММА 50

```

1. 'ALGO1:GAUSS,KOCH,MAX,NORM,РАНЖ,
2. 'BEGIN' 'INTEGER' Н,М,М1,Н1,Р,РМ,МН,Н1,Т,Ж,К,С,Р,Р,Т,
3. 'REAL' XS, Ж,М,РЕАЛН,Ж,М,МН,Н1,РМ.,
4. 'PRINT', //,15ИСХОДНЫЕ ДАННЫЕ,/,'6,15)', 'N,М,М1,МН,Н1,РМ.,
5. 'BEGIN' 'ARRAY' XI//1:M+M1,1:N//',
6. 'INTEGER', 'ARRAY' ТАУ//1:M+M1//, 'BOOLEAN' RR,0000,
7. RR:='FALSE', READXI,ТАУ,Ч0),S:=M+M1,,T:=ТАУ//1//,,P:=S+T,,,
8. 'FOR' I:=2 STEP 1 UNTIL S DO 'BEGIN' IF T LT ТАУ//1//,
9. 'THEN' Т:=ТАУ//1//, P:=P+ТАУ//1//END,,M1:=Н-Т-1,,IF RR THEN
10. 'GOTO' PP,,FOR J:=1 STEP 1 UNTIL S DO 'BEGIN' XS:=0,,,
11. 'FOR' I:=1 STEP 1 UNTIL N DO XS:=XS+XI//J,I//,,XS:=XS/N,,,
12. 'FOR' I:=1 STEP 1 UNTIL N DO 'XI//J,I//':=XI//J,I//XS,,,
13. 'PRINT', '(12ЧИСЛОВЫЕ ПАРАМ.,13,6Х,12СРЕДН.ЗНАЧ. Е12.4)',,
14. J,XS,END,,PP:'BEGIN' 'ARRAY' XI//1:P,1:N1//,

```

```

15. K1(1:R1),KX(1:S1),,,REAL'0.0,I1,T1,TX,YB,C,C1,YM,OTK,AJ.,,
16. ,INTEGER'M,E1,P1,Y1,B1,PY,I,NM,NDM,NG,P,M,J,
17. ,BEG IN'ARRAY'AX(1:N/),,,FOR J:=1 STEP 1 UNTIL S DO'
18. ,REC IN'FOR I:=1 STEP 1 UNTIL N DO'AK(I):=ABS(X(I,1)),,,
19. ,NORM(AX,N,XS),,IX(J,1):=XS,,FOR I:=1 STEP 1 UNTIL N DO'
20. ,XI(J,I):=XI(J,I)/XS END'END',,,OCOC:=TRUE',,
21. ,PRINT'',//,27НУРАБЕРН СОСТАВИЛ ЧМР',,,,
22. ,IF'OCOC'THEN'BEGIN'K:=0,,PRINT'(8ИЗНАЧЕНИЙ)',,END',,IF'
23. ,NOT'OCOC'THEN'BEGIN'K:=1,,PRINT',(10ИЗПАВЕНИЙ)',,END',,
24. ,FOR J:=1 STEP 1 UNTIL M DO'FOR I:=1 STEP 1 UNTIL N1 DO'
25. ,BXK IN'R:=T+1.,Y(J,I,1):=XI(J,R+1)-XI(J,R) F'END',,P:=0.,,
26. ,FOR J:=1 STEP 1 UNTIL S DO'FOR K:=0 STEP 1 UNTIL TAY(J,/)
27. ,BX'BEGIN'P:=P+1.R:=T-K,,FOR I:=1 STEP 1 UNTIL N1 DO'
28. X(J,P,1):=XI(J,I+R)'END',,BEGIN'ARRAY'XM(1:M,1:P)/,,,
29. ,INTEGER'ARRAY'XPL(1:M,1:P/),TX(1:M/),
30. ,INTEGER'ARRAY'TAY,,INTEGER AM,,BEGIN' INTEGER'I,J,N,M,,,
31. ,PROCEDURE'YM(A,B,P,C,M,O),,ARRAY'A,B,C,,INTEGER'P,M,O,,,
32. ,INTEGER'ARRAY'YM(1:N,1:N/,,INTEGER'S,,FOR I:=1 STEP 1
33. ,BEGIN' INTEGER'I,J,K,,FOR J:=1 STEP 1 UNTIL M DO'FOR I:=1
34. ,STEP 1 UNTIL O DO'C(J,I):=O,,P1:I:=1.,P2:'IP' A(J,1)/
35. ,=0'THEN' GOTO'P3.,,FOR R:=1 STEP 1 UNTIL Q DO'IF'B(1,K)>NR'
36. O'THEN C(J,K):=C(J,K)+A(J,I)*B(I,K),,P3:I:=I+1.,IF'
37. I'LE'P'THEN' GOTO'P2.,J:=J+1.,IF'J'LE'M'THEN' GOTO'P1'END',,
38. ,PROCEDURE'SLED(A,N,P),,ARRAY'A,,INTEGER'N,,REAL'P,,BEGIN'
39. ,INTEGER'S,P:=0.,FOR S:=1 STEP 1 UNTIL N DO'P:=P+(S,S)/
40. ,END',,PROCEDURE'SOPR(A,N,P,B),,ARRAY'A,B,,INTEGER'N,,,
41. ,REAL'P,,BEGIN' INTEGER'S,K,,FOR S:=1 STEP 1 UNTIL N DO'
42. ,FOR K:=1 STEP 1 UNTIL N DO'B(S,K):=( IF'K=S'THEN A(S,K) +
43. ,P'ELSE A(S,K))'END',,PROCEDURE'LAMDA(A,N),,ARRAY'A,,,
44. ,INTEGER'N,,BEGIN' ARRAY'P(0:2*N+1/),,,INTEGER'K,,BOOLEAN'
45. G,R,O,P(2*N/):=1.,P(2*N+1):=0.,SLED(A,N,P(2*N-2/)),,P(2
46. *N-2/):=P(2*N-1/):=0.,SOPR(A,N,P(2*N-2/),B),,FOR K:=2*N-4
47. ,STEP 1 UNTIL O DO'BEGIN' P(1:K+1):=0.,YM(A,B,N,A1,N,W),,
48. SLED(A,N,P(K/)),,P(K/):=2*P(K/)(K-2*N),,IF'K'NE'0'THEN'
49. SOPR(A,N,P(K/),B)'END',END',G:=FALSE',,R:=TRUE',,O:=
50. ,FALSE',,PRINT',(14НН03Ф. ПОЛНОМА/,EL2.5),,FOR I:=2*N

```

51. 'STEP 1 UNTIL 0 DO'P(1/),,,BEGIN' ARRAY'KX(1:2\*N/),,REAL'
52. Z,Z1,Z2..,PROCEDURE'ARCSIN(T),,'REAL'T,,BEG IN' REAL'IM..IM:=
53. 3.14159265359.,IF'ABS(T)'LT 1'THEN' ARCSIN:=ARCTAN/Y/SORT'1-
54. T\*2/2)'ELSE'IF'T'LT 0'THEN'ARCSIN:=-IM/2'ELSE'ARCSIN:=IM/2
55. 'END',,HOM(N,G,R,O,5,P,KX),,PRINT'',(КОНФОРМ (Z)=)',,
56. 'PRINT'',(E14.4),,KX,,FOR K:=1 STEP 1 UNTIL N DO'BEGIN'
57. Z1:=K/(2\*I-1/),,Z2:=K/(2\*I/),,Z:=SQRT(Z1\*Z1-Z2\*Z2),,
58. 'IF'Z'GE' 00001'THEN' BEGIN'KX(2\*K-1/):=LN(Z),,KX(2\*K/):=
59. ARCSIN(Z2/Z)'END',ELSE'BEGIN'KX(2\*K-1/):=11.,KX(2\*K/):=0
60. ,END',END',,PRINT'',//,10КОФОРМ (S),,/6E14.4),,KX'END',
61. 'END',N:=AM,,FOR'J:=1 STEP 1 UNTIL M DO'N:=N+TAY(J,1),,
62. ,BEG IN'ARRAY'X(1:N,1:N/,,INTEGER'S,,FOR I:=1 STEP 1
63. ,UNTIL N DO'FOR J:=1 STEP 1 UNTIL N DO'X(J,I):=0.,,FOR'
64. I:=1 STEP 1 UNTIL N-1 DO'X(J+1,I):=1.,S:=0.,,FOR J:=1 STEP
65. 1 UNTIL M DO'BEGIN' IF'J'NE'1'THEN'X(J+S,J+S-1/):=0.,,FOR'
66. I:=1 STEP 1 UNTIL M DO'IF'XPL(J,I)/NE'0'THEN'X(J+S,XPL
67. (J,I)/):=XM(J,I),,IF'OCOC'THEN'GOTO'IP..I:=J+S..X(M,W)
68. :=X(M,W)+1..II:S:=S\*TAY(J,1)'END',,LAMBDA(X,N)'END',END'
69. ,PROCEDURE'EX(M),,'INTEGER'M,,BEGIN' S:=1..M1:=W:=TR(1/)+1.,
70. HAY:'IP'TX(S/):=0'THEN'GOTO'WAT..,FOR J:=1 STEP 1 UNTIL M DO'
71. ,FOR I:=1 STEP 1 UNTIL NI DO'IP'M1'LT XPL(J,I)'THEN'
72. ,GOTO'WAT..M1:=M1-1..TX(S/):=TD(S/)-1.,GOTO'WAT..M1:=TR(1/)+1.,
73. M'THEN'GOTO'KOH..S:=S-1..W:=W+TR(S/)-1..M1:=M1..,GOTO'HAY.,,
74. KOH:'END',,PROCEDURE'RENK(M,TAY,TH),,'INTEGER'M,,INTEGER'
75. ,ARRAY'TH,,BEGIN' INTEGER'ARRAY'NM(1:P/),,K:=S:=0.,
76. ,FOR J:=1 STEP 1 UNTIL M DO'FOR I:=0 STEP 1 UNTIL TAY(J,1)
77. ,DO'BEGIN' S:=S+1..IF'I'LE'TH(J,1)'THEN'K:=K+1..NM(S/):=K
78. ,END',,FOR J:=1 STEP 1 UNTIL M DO'FOR I:=1 STEP 1
79. ,UNTIL NM DO'IP'XPL(J,I)'NE'0'THEN'XPL(J,I):=
80. NM('XPL(J,I)'),,FOR I:=1 STEP 1 UNTIL M DO'BEGIN'
81. TX(J,1):=0..XPL(J,1):=0'END',END',NM:=0.,BM:=40(0/0/),
82. XM(J,1):=0..XPL(J,1):=0'END',END',NM:=0.,BM:=40(0/0/),
83. ,FOR I:=1 STEP 1 UNTIL PM DO'IF'BM'LT 40(V/I)'THEN'BM:=
84. 40(V/I),,BEG IN'ARRAY'X,Y,V(1:B1,1:N1/),K01,K02(1:3/),
85. OM(1:B1/),IM1,IM2(1:3,1:4/),K03(1:B4,1:3,1:PM/),CT,
86. CTP0(0:P/),,INTEGER'ARRAY'HK(1:B1,1:3,1:PM/),E(1:2/),

```

87. NCC(1:B1/),N11,N12(1:W/),N01,N02(1:N1-W/),ND(1:P/),..
88. 'PROCEDURE'NM(M,XH,P,T/),,'ARRAY'XH,IM,,,'INTEGER'P,T.,
89. 'BEGIN' 'INTEGER'S,K,R,,S:=2;R:=J;B:=HM(S:1/);=HM(1,S/);=
90. XH(R,P+1,T/),,HM(S,S/):=XH(R,R,T/),,HM(S,4/):=
91. XH(R,P2,T/),,S:=S+1,,R:=I,, 'IF'S=3'THEN' GOTO'B.,HM(2,3/):=
92. HM(3,2/):=XH(J,I,T/)'END', 'PROCEDURE'OK(N1,N0),,'INTEGER'
93. 'ARRAY'NH,N0,,,'BEGIN' 'INTEGER'I,J,Z,,Z:=0..I:=1..NH:'FOR'J:=1
94. 'STEP'1'UNTIL'P'DO'IP'P:=XH(J/)'THEN' GOTO'M.,Z:=Z+1..
95. NO(Z/):=I,,M:I:=I+1., 'IF'I<=N1'THEN' GOTO'N END', YP:=1.,
96. EO:P2:=HO(V0/),,P1:=q0(V1/),, 'FOR'I:=1'STEP'1'UNTIL'BM'DO',
97. 'FOR'J:=1'STEP'1'UNTIL'PM'DO'NH(1,1,J/):=0., 'PRINT',
98. '(15HH0MEP YPAHEHHR,15)',YP., 'IF'P2=P'THEN' BEGIN' 'FOR'I:=1
99. 'STEP'1'UNTIL'P'DO'ND(1/):=P-I+1., 'GOTO'WE1'END',

100. 'BEGIN' 'ARRAY'D2(1:P/),,'FOR'J:=1'STEP'1'UNTIL'P'DO' 'BEGIN'
101. D2(J/):=0., 'FOR'I:=1'STEP'1'UNTIL'N1'DO'D2(J/):=D2(J/):=
102. D2(J/)+XH(J,I)*Y(XP,I/),,D2(J/):=BS(D2(1/))'END', ..
103. PAHE(D2,ND,P)'END', WEI:'FOR'J:=1'STEP'1'UNTIL'P2'DO',
104. 'FOR'I:=1'STEP'1'UNTIL'N1'DO'XH(J,I/):=X(ND(P-J+1/),I.,
105. NM:=2.,PH:ER=0., 'BEGIN' 'ARRAY'D2(1:N1/),,'INTEGER'
106. 'ARRAY'ND(1:N1/),,'FOR'I:=1'STEP'1'UNTIL'N1'DO' 'BEGIN'
107. D2(I/):=0., 'FOR'J:=1'STEP'1'UNTIL'P2'DO'D2(I/):=D2(I/)+
108. XH(J,I/)*X(1:N1/),,ND(I/),,NM(1:N1/),,FOR'I:=1'STEP'1'UNTIL'P1
109. 'DO' 'BEGIN'NM(1/I/):=ND(I/),,NM2(1/I/):=ND(I+1/I/)'END', ..
110. OM(N11,N01),,OM(N12,N02)'END', 'IF'PH=1'THEN', 'BEGIN' 'ARRAY',
111. AY(1:H1/), 'FOR'I:=1'STEP'1'UNTIL'N1'DO'AY(I/):=
112. ABSY(YP,I/),,MAXAY,N1,W1,T)'END', 'BEGIN' 'ARRAY'
113. XH(1:P2,1:P2+2,1:2/),,R:=N1-H1..S:=P2+2.,W:=P2+1.,
114. 'FOR'J:=1'STEP'1'UNTIL'P2'DO' 'FOR'I:=J'STEP'1'UNTIL'S'DO'
115. 'BEGIN'XH(J,I/),,MAXAY,N1,W1,T)'END', 'BEGIN' 'ARRAY'
116. R'DO' 'BEGIN' E(1/):=H01(R/),,E(2/):=H02(R/),
117. 'IF'I=H1'THEN' 'FOR'T:=1,2'DO'XH(J,I,T/):=XH(J,I,T/)+
118. XH(J,E(T/)),, 'IF'I=S'THEN' 'FOR'T:=1'STEP'1'UNTIL'2'DO'
119. XH(J,I,T/):=XH(J,I,T)+XH(J,B(VT/))*Y(XP,E(VT/)),,
120. 'IF'(VNE M)'AND'(VNE S)THEN' 'FOR'T:=1,2'DO'
121. XH(J,I,T/):=XH(J,I,T)+XH(1,B(VT/))'END', 'END', ..
122. XS:=XM:=0., 'FOR'I:=1'STEP'1'UNTIL'R'DO' 'BEGIN'

123. XS:=XS+Y(J,YP,N01(VT/)),,E1:=XM+Y(J,YP,N02(VT/))'END', ..
124. ::=1.,J:=0.,T:=J+1.,T:=J+1.,E(HM(1,1,V/):=HM2(1,1,V/):=R.,
125. HM(1,4/):=XS.,HM2(1,4/):=JM.,NM(HM1,XH,P2,1)..,
126. HM(HM2,XH,P2,2),,GAUSS(HM1,3,K01,H),,GAUSS(HM2,3,K02,H),,
127. C:=C1:=0., 'FOR'K:=1,2,3'DO' 'BEGIN' C:=C+(K01(K/V)-K02(K/V))*2.,
128. C1:=C1+K01(VK/)*2+K02(VK/)**2'E1D.,C:=C/C1., 'IF'W'GT'P1
129. 'THEN' 'GOTO'M.,JM(W/):=C., 'FOR'K:=1,2,3'DO'K03(M,K,PH/):=
130. K01(VK/),,NM(W/2,PH/):=J.,NM(W/3,PH/):=I.,
131. 'IF'W=P1'THEN'PAHE(CM,NC,P1.),,W=W+1., 'GOTO'M.,W:
132. 'IF'C'GT'CM(NC(P1/))':=C., 'FOR'K:=1,2,3'DO',
133. K00(NC(P1/),K,PH/):=K01(VK/),
134. NM(NC(P1/),2,PH/):=I.,NM(NC(P1/),3,PH/):=I.,PAHE(CM,NC,P1),,
135. 'GOTO'M.,M:ER:=ER+1..M:I:=I+1., 'IF'I'L'E'P2'THEN' GOTO'L.,
136. 'IF'J'LT'P2-1'THEN' 'GOTO'T., 'FOR'S:=1'STEP'1'UNTIL'P1'DO'
137. 'BEGIN'OM:=0., 'FOR'J:=1'STEP'1'UNTIL'N1'DO' 'BEGIN'YB:=
138. K03(S,1,PH/),, 'FOR'Y:=2,3'DO'YB=YB+K03(S,1,PH/)*
139. XX(NK(S,I,PH/),J/),,OM:=OM+(Y(XP,J/)-YB)**2'END', ..
140. OM:=SORT(OM/N1)/YM., 'IF'S=1'THEN'TX:=TJ:=OM., 'IF'NC(1/):S
141. 'THEN' 'BEGIN'AT:=OM., 'IF'PH=1'THEN' 'BEGIN' OM:=OM.,
142. PH=1..NM:=S'END', 'IF'OM>0'THEN' 'BEGIN' OM:=OM., PH:=PH.,,
143. NM:=S'END', 'END', 'IF'OM'LT'TJ'THEN'TJ:=OM., 'IF'OM'GT'TX
144. 'THEN' TX:=OM'END', 'PRINT', '(//, 4НРДА, 13, 8, 8ИМБЯНО
145. ВИОЛСАНН, 14ЮХ, 11ERROR GAUSS, 16)', PH, P1, ER.,
146. 'PRINT', '(10X, 16ШOKA3. CMEL, JV4UL, E12.F, 6НХУДШ., E12.6)',,
147. OM(NC(1/)),,OM(NC(P1/)),, 'PRINT', '(10X, 14НОЧЕКА НА П/П.
148. ,7Н ЛУЧШ., 2E12.4, 6НХУДШ., E12.4)', T1, AL, TX, SH, LIM:=LIM*2.,
149. 'FOR'J:=1'STEP'1'UNTIL'P1'DO' 'FOR'I:=1'STEP'1'UNTIL'N1
150. 'DO' 'BEGIN'YM(J,I/):=K03(J,1,PH/),, 'FOR'S:=2,3'DO'
151. YY(J,I/):=YY(J,I)+XX(NK(J,S,PH/),1)*K03(J,S,PH/)'END'
152. ., 'FOR'J:=1'STEP'1'UNTIL'P1'DO' 'FOR'I:=1'STEP'1'UNTIL'N1
153. 'DO'XH(J,I/):=yy(J,I),, 'IF'PH=PM'THEN' 'GOTO'J'END', ..
154. PH:=PH+1.,P2:=P1.,P1:=q0(PH/),, 'GOTO'PHD.,L:
155. PH:=PH,NG:=NM., 'PRINT', '(10X, 13НВРАЮ РИЛОВ, 15)', PH, JIDM
156. ::=HM-1., 'FOR'I:=J'STEP'1'UNTIL'P'DO'CTP(I/):=CT(I/):=0.,
157. 'BEGIN' 'ARRAY'A,AD(1:IDM),,'INTEGER'ARRAY'AN,AD(1:IDM),,
158. 'FOR'I:=1'STEP'1'UNTIL'IDM'DO' 'BEGIN'

```

```

159.  $\text{AN}(\text{V}_1) := \text{AD}(\text{V}_1) := 0$ ,  $\text{AN}(\text{V}_1) := \text{AND}(\text{V}_1) := 0$  'END' ,
160. 'FOR' T:=1,2,3'DO', 'BEGIN'  $\text{A}(\text{V}_1) := \text{KOC}(\text{ANG}, \text{T}, \text{PV}_1)$  , ,
161.  $\text{AN}(\text{V}_1) := \text{MK}(\text{V}_1, \text{ANG}, \text{T}, \text{PV}_1)$  'END' , 'IF'  $\text{PV}_1 = 1$  'THEN' 'GOTO'  $\text{NN}$ ,  $\text{III} := 1$  ,
162.  $\text{AD}(\text{V}_1) := \text{A}(\text{V}_1)$ ,  $\text{MII} := 1$  ,
163. 'FOR' T:=2'STEP'1'UNTIL'  $\text{NM}$ 'DO', 'IF'  $\text{A}(\text{V}_1) \neq 0$  'THEN'
164. 'BEGIN'  $\text{IN}'$ , 'IF'  $\text{AN}(\text{V}_1) = 0$  'THEN' 'BEGIN'  $\text{IN}'$  : =  $\text{III} + 1$ ,  $\text{AD}(\text{V}_1) := \text{A}(\text{V}_1)$  , ,
165.  $\text{AND}(\text{V}_1) := 0$  'END' 'ELSE', 'BEGIN' 'FOR' S:=1,2,3'DO', 'BEGIN'
166.  $\text{AD}(\text{V}_1) := \text{KOC}(\text{ANG}, \text{V}_1, \text{S}, \text{PV}_1) * \text{A}(\text{V}_1)$  ,  $\text{AND}(\text{V}_1) :=$ 
167.  $\text{MK}(\text{V}_1, \text{ANG}, \text{V}_1, \text{S}, \text{PV}_1)$  'END' ,  $\text{III} := \text{III} + 3$  , 'END' 'END' ,
168. 'FOR' I:=2'STEP'1'UNTIL'  $\text{NM}$ 'DO', 'BEGIN'  $\text{A}(\text{V}_1) := \text{MK}(\text{V}_1)$  , ,
169.  $\text{AN}(\text{V}_1) := \text{AN}(\text{V}_1)$  , 'END' , 'PI' :=  $\text{PI} - 1$  , 'IF'  $\text{PI} \neq 1$  , 'THEN' 'GOTO'  $\text{MM}$  ,
170.  $\text{NN} := \text{PI}$ , 'FOR' I:=1'STEP'1'UNTIL'  $\text{NM}$ 'DO', 'BEGIN'  $\text{S} := \text{AN}(\text{V}_1)$  , ,
171.  $\text{CT}(\text{V}_1) := \text{CT}(\text{V}_1) * \text{A}(\text{V}_1)$  'END' , 'FOR' I:=1'STEP'1'UNTIL'  $\text{PV}$ 'DO'
172. 'STROK(\text{AD}(\text{V}_1)), 'CT(\text{V}_1)' 'END' , 'STROK(0) := \text{CT}(0) * \text{IK}(\text{NP})
173. , 'S := 0, 'PRINT' , '(12ИКОЭФФИШЕНТЫ)' ,
174. 'PRINT' , '(14L.5)', 'STROK(0)' , 'FOR' J:=1'STEP'1'UNTIL'  $\text{M} + \text{M}_1$ 'DO'
175. 'FOR' I:=1'STEP'1'UNTIL'  $\text{TA}(\text{V}_1)$ 'DO', 'BEGIN'  $\text{S} := \text{S} + 1$  ,
176. 'IF' 'STROK(S)'  $\neq 0$  'THEN' 'BEGIN'  $\text{CTROK}(S) := \text{STROK}(S)$  *
177.  $\text{IK}(\text{V}_1) := \text{IK}(\text{V}_1)$ , 'IF' 'LE'  $\neq 1$  'THEN' 'BEGIN'  $\text{H} := \text{H} + 1$  ,
178. 'IF' 'TK(V_1/J)'  $\neq \text{I}$ .  $\text{XK}(\text{V}_1, \text{H}) := \text{STROK}(S)$  ,
179.  $\text{XPL}(\text{V}_1, \text{H}) := \text{S}$  'END' , 'PRINT' , '(14, 3X, 5НСДВИГ, 14, 5Х, R14, 5)' ,
180.  $\text{J}, \text{I}, \text{STROK}(S)$  'END' 'END' , 'PRINT' , '()' ,
181. 'IF' 'MI'  $\neq 1$  'THEN' 'MI := 1',  $\text{UP} := \text{UP} + 1$  , 'IF' 'UP'  $\leq \text{M}' 'THEN' 'GOTO' EQ
182. 'END' , 'RENM(M, TA, TI)', 'PRINT' , '(20X, 1ЭСИСТЕМА, //)' ,
183. 'STAN(M, TA)', 'IF' 'MI'  $\neq 1$  'THEN' 'GOTO' OK, 'R := MII' ,
184. 'FOR' J:=1'STEP'1'UNTIL'  $\text{MI}$ 'DO', 'R := R + TK(V_1/J) ,
185. 'FOR' J:=1'STEP'1'UNTIL'  $\text{MI}$ 'DO', 'FOR' I:=1'STEP'1'UNTIL'
186.  $\text{NM}$ 'DO', 'IF' 'R'  $\leq \text{XPL}(J, 1)$  'THEN' 'XPL(J, 1) := XK(V_1, 1)' := 0 , ,
187. 'FOR' J:=1'STEP'1'UNTIL'  $\text{MI}$ 'DO', 'TD(K/J) := Tk(V_1/J)' , 'EX(MII)' ,
188. 'RENM(MI, TI, TD)', 'PRINT' , '(18X, 1ОШЮДСИСТЕМА, 5Х, 1И1)' ,
189. 'CTB(MI, TI)', 'FOR' I:=1'STEP'1'UNTIL'  $\text{MI}$ 'DO', 'BEGIN'
190.  $\text{TK}(J/J) := \text{TK}(\text{MI} + \text{J})$ , 'TD(J/J) := Tk(V_1/J)' , 'END' , 'K := I := 0 , ,
191. 'FOR' J:=MI+1'STEP'1'UNTIL'  $\text{MI}$ 'DO', 'BEGIN'
192.  $\text{I} := \text{I} + 1$  'UNTIL'  $\text{MI}$ 'DO', 'BEGIN'  $\text{XK}(\text{V}_1, \text{I}) := \text{XPL}(V_1, I)$  := 0 , ,
193. 'IF' 'XPL(V_1, I)'  $\geq \text{TK}(J, I)$  'THEN' 'BEGIN'  $\text{S} := \text{I} + 1$  ,
194. 'XPL(V_1, S) := XPL(V_1, I) - R, 'XK(V_1, S) := XK(V_1, I)' , 'END' 'END' ,$ 
```

```

195. 'TK' 'LT' 'S' 'THEN' 'S := END' , 'NM := III', 'EX(M-MI)' , 'REN(M-MI, TI, TD)' ,
196. 'PRINT' '(18X, 1ОШЮДСИСТЕМА, 5Х, 1И2)' , 'STAR(M-MI, TI, TD)' , 'OK' :
197. 'END' 'END' 'END' 'END' 'END' 'EOF'

```

### 3. ПРОГРАММА УПРАВЛЕНИЯ С ОПТИМИЗАЦИЕЙ ПРОГНОЗА ЛИНЕЙНОЙ ДИСКРЕТНОЙ СИСТЕМОЙ С ЗАПАЗДЫВАНИЕМ

Предназначена для расчета оптимального управления многомерным дискретным объектом, описываемым разностными линейными уравнениями с запаздывающими аргументами, при заданном критерии оптимизации (прогр. 51) [34]. Язык программирования АЛГОЛ, ЭВМ БЭСМ-б

**Используемые подпрограммы.** ОРГ(M), НОД

**Метод.** Динамика объекта описывается следующей системой разностных уравнений:

$$\bar{X}_{k+j} = \sum_{v=0}^N A^v \bar{X}_{k+v-u-1} + \sum_{v=0}^N B^v u_{k+v-u-1} + \sum_{v=0}^{N_A} C^v \bar{u}_{k+v-u-1}, \quad v = \overline{1, N}. \quad (5.8)$$

При известных начальных и предналичных условиях

$$\bar{X}_{k-v} = \bar{X}_{k-v}^0, \quad v = 0, 1, 2, \dots, N_X; \quad (5.9)$$

$$\bar{U}_{k-v} = \bar{U}_{k-v}^0, \quad v = 0, 1, 2, \dots, N_U; \quad (5.10)$$

$$\bar{h}_{k-v} = \bar{h}_{k-v}, \quad v = 0, 1, 2, \dots, N_h, \quad (5.11)$$

где  $\bar{X}^T = |X^1 \dots X^n|$ ,  $\bar{U}^T = |U_1 \dots U^n|$ ,  $\bar{\lambda}^T = |\lambda_1 \dots \lambda_n|$  — векторы выходных величин, управлений и измеряемых (протогизуемых) возмущений разности  $n \times n$ ,  $n \times u$  и  $n \times v$  соответственно;  $A^v$ ,  $B^v$  и  $C^v$  — матрицы коэффициентов разности  $n \times n$ ,  $n \times u$  и  $n \times v$  соответственно;  $k$  — номер текущего такта управления относительно текущего;  $v$  — номер будущего (расчетываемого) такта управления или возмущения;  $u$  — значение запаздывания по выходной величине, управлению или возмущению;  $N_X$ ,  $N_U$ ,  $N_h$  — максимальные значения запаздываний по возможным величинам, управлением и возмущениям соответственно;  $N$  — число тактов оптимизации.

Заданы ограничения на выходные величины и управления объекта (5.8):

$$\bar{X}^{\min} \leq \bar{X}_{k+v} \leq \bar{X}^{\max}, \quad v = \overline{1, N}. \quad (5.12)$$

Критерий функционирования системы управления в общем случае может зависеть от всех выходных величин и управлений, в данной программе реализован квадратичный критерий вида

$$f = \sum_{v=k+1}^{k+N} \alpha_v (\bar{P}_v, Q_v, \bar{Q}_v), \quad (5.14)$$

где

$$\Phi_v^t = 1 - \frac{X_v^t}{\Psi_v^t}, \quad t = \overline{1, n} \quad (5.15)$$

— относительная ошибка регулирования;  $\bar{\Psi}_j$  — заданный вектор выходных величин объекта на  $j$ -м этапе управления;  $\Psi$ . В выражении (5.14)  $\alpha_{\Psi}$  — весовой коэффициент, учитывающий значимость отдельных тактов управления;  $Q_{\Psi}$  — положительная определенная матрица весовых коэффициентов разности  $n \times n$ , учитывающих значимость заданного такта отдельных выходных величин объекта.

Данная программа предназначена для решения задач как с фиксированым интервалом оптимизации (при заданном  $k$ ), так и со скользящим (при  $N = \text{const}$  и переменном  $k$ ), а также для получения совокупности управлений на интервале  $U_{k+\gamma}$ ,  $\gamma = 1, N$ , обеспечивающей экстремум функционалу (5.14) при учете ограничений (5.12) — (5.13). В случае скользящей оптимизации для реализации на объекте используется только первый член этой совокупности  $U = \bar{U}_{k+1}$ .

Метод решения заключается в следении задач (5.8), (5.15) к соответствующей задаче математического программирования. Уравнения объекта (5.10) — (5.12) можно рассматривать как некоторые линейные ограничения в форме равенств

$$\bar{h}(\bar{X}) = 0, \quad \bar{h}^T = [h^1 \dots h^{n \times N}], \quad (5.16)$$

а двусторонние ограничения (5.12), (5.13) — нелинейные ограничения в форме односторонних неравенств

$$\tilde{g}(\bar{X}) \leqslant 0, \quad \bar{g}^T = |g^1 \dots g^{(n+\gamma)N}|. \quad (5.17)$$

Причем вектор переменных  $\bar{X}$  представляет собой совокупность искомых управляемых воздействий и выходных величин объекта:

$$\bar{X}^T = [\bar{X}_{k+1}^T, \bar{U}_{k+1}^T, \dots, \bar{X}_{k+\gamma}^T, \bar{U}_{k+1}^T]. \quad (5.18)$$

Задачу оптимизации (5.14) для каждого ее значения можно рассматривать как задачу оптимизации некоторой функции

$$f = f(\bar{X}) \quad (5.19)$$

при ограничениях (5.16), (5.17), которая решается с помощью программы ОРТИМ1.

#### Описание параметров.

*Базовые:*

$M$  — общее число переменных ( $n + \gamma + l$ );  $V2$  — ( $V2 = \text{ТАУ} + 1$ ), где ТАУ — максимальное значение запаздывания для аргументов модели (в программе принято  $N_X = N_U = N_A = \text{ТАУ}$ );

$L2$  — число тактов оптимизации;

$LL$  — количество выполняемых тактов управления;

$L3$  — число производимых переменных;

$UV$  — число управляющих воздействий;

$L6$  — число измеряемых возмущений (аргумент  $T$  — время в это число не входит);

$L5$  — вспомогательный параметр ( $L5 = M - L3 = L6 + 1 + UV + YT$ );  
 $YT$  — кратчайший ( $YT = 1$ ) — есть аргумент  $T$ ;  $YT = 0$  — нет  $T$ );  
 $DG$  — периодичность (по числу итераций) вывода на печать промежуточных результатов поиска экстремума ( $+DG$  — сокращенная информация,  $-DG$  — полная информация);

$CQ$  — максимальное количество итераций при поиске экс-

тремума;  
 $G$  — соответствует коэффициенту  $\alpha$  (см. описание ОРТИМ1);  
 $(+G$  — поиск максимума,  $-G$  — поиск минимума);

$СШ$  — точность выбора шага в направлении вектора сдвига (ограничение на минимальное приращение);

$ТУ$  — начальное значение пробного шага;

$\mathcal{E}$  — точность соблюдения ограничений;

$E1, E$  — установки для останова процедуры поиска экстремума (см. описание ОРТИМ1);

$ПЛ[1 : Z1, 1 : 2, 1 : L3]$  — массив коэффициентов модели (5.8),  $Z1 = (M - YT) \times V2 + YT + YU + 1$  вычисляется. Последовательно вводится коэффициент каждого уравнения (включая пусты) в порядке, соответствующем уравнению (5.8): на первом месте — свободный член, на последнем —

коэффициент при  $U_{+1}$ ; в первом же порядке, в каком расположены

коэффициенты в ПЛ;

$IX[1 : L2+LL, 1 : L5]$  — массив значений измеряемых возмущений ( $\bar{A}$ ) на  $L2 + LL$  тактах вперед;

$ОГ[1 : 2, 1 : L4]$  — массив ограничений на векторы  $\bar{X}$  и  $\bar{U}$ . Порядок ввода:

$X_{\min}^1, X_{\max}^1, \dots, X_{\min}^n, X_{\max}^n, U_{\min}^1, U_{\max}^1$ ;

$MB1[1 : L3]$  — массив весовых коэффициентов  $\alpha_{\Psi}$  (5.14);

$MB2[1 : L3, 1 : L2]$  — массив значений весовой матрицы  $Q_{\Psi}$  (5.14);

$ХС, ХОС[1 : Z1-1]$  — основной и дублирующий массивы значений началь-ных условий (5.10), (5.11). В них вводятся значения

переменных в том же порядке, в каком расположены

значение оптимизируемого функционала (5.15);

$X[1 : NO]$  — массив значений протозируемых переменных и управлений на  $L2$  та-ктах оптимизации (в начале работы сюда записываются начальные приближения из массива  $ХОС$ ),  $NO = (L3 + YT) \times L2$ .

Последовательность работы программы. Выполнение работы сюда записываются операции, в программе начинается со строки 84, где в цикле выполняется  $LL$  этапа оптимизации. На каждом этапе оптимизации предварительно заполняется массив  $X$  (на все  $L2$  такта одинаково) из массива начальных условий  $ХОС1$ . Далее обращением к процедуре ОРТИМ1 находится оптимальное значение  $F$  и  $X$ . Производится сви-дантих на один шаг в массиве  $ХОС1$  и  $IX$  и далее процедура вычислений повторяется ( $LL$  раз). ОРТИМ1 обращается в свою очередь к подпрограммам EQUATI и JACOB1. В EQUATI вычисляются значения переменных по формулам (5.16), (5.17) и значение функционала (5.14). При этом используется процедура PREDICTION, в которой вычисляется значение производимых переменных. В процедуре JACOB1 вычисляется матрица частных производных системы опра-ничений (5.16), (5.17) и градиент функционала по некотору переменных (5.18).

**Пример 5.** Объект описывается разностным уравнением с запаздывающими аргументами

$$\dot{X}_{+1} = 5 + X_0 - X_{-1} + X_{-2} - X_{10} - \lambda_{1,-1} + \lambda_{1,-2} + \lambda_{2,-1} - U_0 + U_{+1}$$

(обозначения те же, что и в формуле (5.8)).

Даны начальные условия:  $X_0 = 1, X_{-1} = 1.5; X_{-2} = 1.2; \lambda_{10} = 2; \lambda_{1,-1} = 3; \lambda_{1,-2} = 4; \lambda_{20} = 7; \lambda_{2,-1} = 3; U_0 = 46.7$ . Требуется обеспечить следующие значе-ния выходной величины:  $X_{+1} = 1; X_{+2} = 1; X_{+3} = 1$ . При этом используется информация об измеряемых возмущениях:  $\lambda_{1,-1} = 3; \lambda_{1,-2} = 1; \lambda_{2,-1} = 5$ .

Применяется следующий режим оптимизации: на каждом этапе управления опти-мизируется функционал, вычисляемый на два такта вперед ( $L2 = 2$ ), и количество этапов управления равно  $LL = 2$ . Заданы ограничения:  $X_{\min} = 0.5, X_{\max} = 2$ ;  $U_{\min} = 10; U_{\max} = 60$ .

Принятые следующие значения установок управляющей карты:  $M = 4; V2 = 3; L2 = 2; LL = 2; L3 = 1; UV = 1; L6 = 2; L5 = 3; YT = 0; DG = 10; CQ = 50$ ;

$G = -1; СШ = 0.005; TU = 1.0; \mathcal{E} = 0.1; \mathcal{D} = 0.0001; E = 0.0001$ ;

**Решение 1 этапа управления:**  $f_1 = 0.000021; X_{+1} = 0.9999; U_{+1} = 39.99$ ;

$X_{+2} = 0.99797; U_{+2} = 29.498$ .

**Решение II этапа управления:**  $f_2 = 0.0000045; X_{+2} = 0.99928; U_{+2} = 29.499$ ;

$X_{+3} = 0.997096; U_{+3} = 21.497$ .

```

1. 'ALGOL':HOA,OPTIM.,*
2. 'BEGIN'INTEGER'LL,I2,I3,I4,I5,I6,YB,YT,V2,IM,NO,NL,Z1,M,N2,
3. TAY,I,J,K,AT,C,O,,REALG,C,W,TY,3,E,E1,F,C,31,,READM,V2,I2,
4. LL,L3,YB,I6,I5,YT),READGT,O,G,C,W,TY,3,J1,E,,,'PRINT'.,,'C//'
5. //,40X,12H 3 A Д А Н 0 ./,40X,13(1H-))'.,,'PRINT'.,,'C//',13,
6. 5X,3HW2=,I2,5X,3HW2=,I2,5X,3HW2=,I2,5X,3HW2=,I2,5X,3HW2=,I2,
7. 5X,3HW2=,I2,5X,3HW2=,I2,5X,3HW2=,I2,5X,3HW2=,I2,5X,3HW2=,I2,
8. 15,YT,,,'PRINT'.,,'C//',3HW2=,I4,5X,3HC0=,I3,5X,2HG=,F10,4,5X,
9. 3HC0=,F10,4,5X,3HW2=,F10,3,5X,2HG=,F10,5,5X,3HA1=,E10,6,5X,
10. 2HG=,F10,6)'.,,GT,CQ,G,CW,TY,9,31,E,NO:=ID:=(I3+YB)*I2,,IM:=
11. (2 L3*YB)*I2,,I4:=I3+YB,,Z1:=(I4-YT)*V2+YT+YB+1,,TAY:=V2-1,
12. 'PRINT'.,,'C//',3HW2=,I3,5X,3HW2=,I3,5X,3HW2=,I3,5X,3HW2=,I3),
13. NO,IM,I4,Z1,'BEGIN'ARRAY'IM(/1:Z1,1:2,1:I3/),XOC,XOC1
14. (/1:Z1-1/).MB1(/1:I3/),MB2(/1:I3,1:I2/),NX(/1:I3,1:I2/),
15. ГУЛ(/1:I3,1:I2/),IX(/1:I2+LL,1:I5/),OR(/1:2,1:I4/),
16. X(/1:IH0/),Y(/1:IM/),NY(/1:IM,1:NO/),NF(/1:NO/);'INTEGER'
17. 'ARRAY'IM(/1:I3,1:I2/),,'PROCEDURE' PREDICTION(M,I5,C,LL,IM,
18. Z1,XOC,PY,I,M2,V2,YT,TAY,IX,YB),,'INTEGER'M,I5,Z1,V2,M2,YT,TAY,
19. YB,,,'ARRAY'IM,XOC,PY,I,IX,,,'INTEGER'ARRAY'IM,,,'REAL'C.,
20. 'BEGIN'INTEGER'R,I,J,,,'FOR'R:=1'STEP'1'UNTIL'M-1'DO',
21. 'BEGIN'C:=IM/(1,IM/(*T,I),T/),,'FOR'I:=1'STEP'1'UNTIL'Z1-1
22. 'DO'C:=C+XOC(/1/*IMK/I+1,IM/(*T,1),T/).PYL(M,N2):=XOC
23. (/T-1)*V2+1)*C'END,,,'FOR'I:=1'STEP'1'UNTIL'M-1'DO,'FOR'
24. J:=TAY'STEP'-1'UNTIL'1'DO,XOC(/J+1*(I-1)*V2/):=XOC(/J+(*I-1)
25. V2/),,'FOR'I:=1'STEP'1'UNTIL'M-1'DO,XOC(/1+(*I-1)*V2/):=PYI
26. (/I,N2/),,'FOR'I:=1'STEP'1'UNTIL'1'DO,XOC(/M-1+I-1)*V2+1):=
27. IX(/N2,I/),,'FOR'I:=1'STEP'1'UNTIL'YB'DO,XOC(/Z1-1-YB+1):=
28. IX(/N2+1,I/)'END,,,'PROCEDURE'EQUAT(X,F,Y),,'ARRAY'X,Y,,,
29. 'REAL'F,,,'BEGIN'INTEGER'I,J,M2,,,'REAL'C,,,'FOR'I:=1'STEP'1
30. 'UNTIL'Z1-1'DO'XOC(/1):=XOC(/I/),,'FOR'I:=1'STEP'1'UNTIL'Z2
31. 'DO,'FOR'J:=1'STEP'1'UNTIL'L4'DO'Y(/J+(*I-1)*L4/):=(X(/J+
32. (*I-1)*L4/-0R(/V,I,J/))*(X(/J+(*I-1)*L4/)-0R(/2,J/)),,F:=0,,,
33. 'FOR'N2:=1'STEP'1'UNTIL'L2'DO,'BEGIN'FOR'I:=1'STEP'1'UNTIL'
34. YB'DO'IX(/M2,15-YB-YT+1):=X(/L4*(M2-1)+L3+1),,,,
35. 'FOR'I:=1'STEP'1'UNTIL'YB'DO'XOC(/Z1-1-YB+1):=X(/L4*(M2-1)+

36. 13+1/),,'PREDICTION(M,L5,C,LL,IM,Z1,XOC,PY,I,M2,V2,YT,TAY,IX,
37. YB),,'FOR'I:=1'STEP'1'UNTIL'L3'DO'Y(/L4*L2+(M2-1)*L2*T/):=
38. NYL(/I,N2/)-X(/(M2-1)*L4+1/),,'FOR'I:=1'STEP'1'UNTIL'L3'DO'
39. F:=F+0.5*MBR(/I/)*MBR(/I,N2/)*(1-X(/J+(M2-1)*L4/)/XX(/I,N2/))
40. **2'END,'END'EOUTT,,,'PROCEDURE'JACOBI(X,NF,NY),,'ARRAY'X,
41. NF,NY,,,'BEGIN'INTEGER'I,J,K,H,S,,,'FOR'I:=1'STEP'1'UNTIL'L2
42. 'DO,'FOR'J:=1'STEP'1'UNTIL'L3'DO'HR/J+(I-1)*L4/):=MBR(/J/)*
43. MBR(/J,1/)*(1-X(/J+(I-1)*L4/))/XX(/J,1/))/XK(/J,1/),,'FOR'
44. I:=1'STEP'1'UNTIL'L2'DO,'FOR'J:=1'STEP'1'UNTIL'L4'DO'
45. NY(/(I-1)*L4+J,(I-1)*L4+J/):=2*X(/(I-1)*L4+J/-6T(/1,J/)-
46. 0T(/2,J/),,'FOR'I:=1'STEP'1'UNTIL'L3'DO'NY(/L2*L4+I,I/):=-1,
47. S:=L2,,,'IF'L2'GT'(2+TAY),THEN'S:=2*TAY,,,'FOR'I:=1'STEP'1
48. 'UNTIL'S=1'DO,'FOR'J:=1'STEP'1'UNTIL'L3'DO,'FOR'K:=1'STEP'1
49. 'UNTIL'L3'DO'NY(/L2*L4+I*L3+J,F/):=TIL(/1+(K-1)*V2+I,1,J/),
50. 'FOR'J:=1'STEP'1'UNTIL'L3'DO'FOR'K:=1'STEP'1'UNTIL'YB'DO'
51. NY(/L2*L4+J,L3+K/):=ILK(/Z1-YB+K,1,J/),,'FOR'I:=1'STEP'1
52. S-1'DO,'FOR'J:=1'STEP'1'UNTIL'L3'DO,'FOR'K:=1'STEP'1'UNTIL'
53. YB'DO'NY(/L2*L4+I*L3+J,L3+K/):=ILK(/1+V2*(L3+L6)+(K-1)*
54. V2+I,1,J/),,'FOR'I:=1'STEP'1'UNTIL'S-1'DO,'FOR'T:=1'STEP'1
55. 'UNTIL'S=H'DO,'FOR'J:=1'STEP'1'UNTIL'L4'DO,'FOR'K:=1'STEP'1
56. 'UNTIL'L3'DO'NY(/L2*L4+(H-1)*L3+K,L4*I-V/):=HY(/L2*L4+
57. (H-1)*L3+K,J/)'END'JACOBI,,,'FOR'I:=1'STEP'1'UNTIL'L3'DO,'FOR'
58. J:=1'STEP'1'UNTIL'Z1'DO'READNL(/J,1,I/),,'
59. PRINT'.,,'C//',25IMACCB КОМФИМЕНТОВ (ИМ),/),,,,'PRINT'
60. '(OE12.4)',,'FOR'J:=1'STEP'1'UNTIL'L3'DO'FOR'J:=1'STEP'1
61. 'UNTIL'Z1'DO'ILK(/J,1,I/),,'FOR'I:=1'STEP'1'UNTIL'Z1'DO,'FOR'
62. J:=1'STEP'1'UNTIL'L3'DO'ILK(/I,2,J/):=ILK(/I,1,J/),,
63. 'FOR'I:=1'STEP'1'UNTIL'L3'DO'ILK(/2+V2*(I-1),2,I/):=ILK(/2+V2*
64. (I-1),2,I/)-1,,READ(XC),,'PRINT'.,,'C//',18НННВАЛЬНЫЕ УСЛОЖН
65. 5IL(XOC),/),,,,'PRINT'.,'(OE12.4)',,'XOC,,,'FOR'I:=1'STEP'1
66. 'UNTIL'L6'DO,'FOR'J:=1'STEP'1'UNTIL'L2+LL'DO'
67. READ(TX(/J,I/)),,'IF'YT'E01'THEN''BEGIN'TX(/1,L5/):=XOC
68. (/Z1-1-YB)*1,,,'FOR'I:=2'STEP'1'UNTIL'L2+LL'DO'IX(/I,L5/):=
69. IX(/I-1,L5/)+1,,,'PRINT'.,,'SUBPAR T,/,),,,,'PRINT'.,'(OE14)'',
70. 'FOR'I:=1'STEP'1'UNTIL'L2+LL'DO'1/(V1,L5/)'END,,,'PRINT'.,
71. 'C//',40ИМЕРНДИЕ ВОЗМОЖЕНИЯ (IX(/1:L2+LL,1:L6/)),/),,,,'

```

```

72. 'PRINT',(10E12.4),,'FOR' I:=1 'STEP' 1 'UNTIL' L6'DO,'FOR' J:=1
73. 'STEP' 1 'UNTIL' L2+LL'DO IX/J,I/,,READ(0T),'PRINT' ,//,
74. 16ИГРАНИЧЕНИЯ (0T),/,, 'PRINT' ,(10E12.4),,'0T,,,'FOR' I:=1
75. 'STEP' 1 'UNTIL' L3'DO,'FOR' J:=1 'STEP' 1 'UNTIL' L2'DO,'BEGIN'
76. MBL(/1/):=MBZ(/1,J/):=0.,XX(/1,J/):=1'END',,MB1(/1/):=1.,
77. MBZ(/1,1/):=1.,MBZ(/1,2/):=1.,XX(/1,1/):=1.,XX(/1,2/):=1.,
78. 'PRINT' ,//(//,24НЗНАЧЕНИЙ ФУНКЦИИ ЦЕЛИ XX,/),,, 'PRINT'
79. ,(E12.4),,XX,'PRINT' ,//(//,19НЕСОВЫЕ КОЭФФ.(MB1),/),,,,
80. 'PRINT' ,(F10.3),,MB1,, 'PRINT' ,(//,19НЕСОВЫЕ КОЭФФ.(MBZ),
81. /),,, 'PRINT' ,(F10.3),,MBZ,, 'FOR' I:=1 'STEP' 1 'UNTIL' Z1-1'DO'
82. XC1(/1/):=XC1(/1/),, 'FOR' I:=1 'STEP' 1 'UNTIL' L3'DO,IM(/I,1/)
83. :=2., 'FOR' K:=1 'STEP' 1 'UNTIL' L1'DO,'BEGIN' , 'FOR' I:=1 'STEP' 1
84. 'UNTIL' L3'DO,'BEGIN' X(/I/):=XC1(/(I-1)*V2+1/),, 'FOR' J:=2
85. 'STEP' 1 'UNTIL' L2'DO,X(/(J-1)*L4+1/):=X(/I/)'END',,
86. 'FOR' I:=1 'STEP' 1 'UNTIL' V2'DO,'BEGIN' K(/I+L3/):=XC1(/Z1-1-VB-
87. 1/),, 'FOR' I:=2 'STEP' 1 'UNTIL' L2'DO,X(/(J-1)*L4+1+L3/):=X(/I+
88. L3/)'END',, 'PRINT' ,//(13,18Н-й этап управления,/,21(Н-)),,
89. K.,OPTIM(X,IM,NO,ND,ML,LT,CQ,C,CS,TU,3,31,E,F,EQUAT,JACOBI,0T,L4),,
90. 'PRINT' ,//(10Х,ТИРЕЛЕНИЕ,/,9Х,10(Н-)),,, 'PRINT'
91. ,,(C,2H1F-,E20.9,/,2HХ=),,F,, 'PRINT' ,(10E12.5),,X,, 'FOR'
92. I:=1 'STEP' 1 'UNTIL' M-UT'DO,'FOR' J:=TAU,STEP,-1 'UNTIL' 1'DO,
93. XC1(/J+1+(I-1)*L2/):=XC1(/J+(I-1)*V2/),, 'FOR' I:=1 'STEP'
94. 1'UNTIL' M-15 'DO' XC1(/1+(I-1)*V2/):=TUX(/1,1/),, 'FOR' I:=1
95. 'STEP' 1 'UNTIL' L5'DO,LC1(/(L3+I-1)*V2+1/):=TUX(/1,1/),, 'FOR'
96. I:=1 'STEP' 1 'UNTIL' UV'DO XC1(/Z1-I-UV+1/):=TUX(/1,L6,I/),
97. 'FOR' I:=1 'STEP' 1 'UNTIL' L2+LL-1'DO,'FOR' J:=1 'STEP' 1 'UNTIL'
98. L5'DO IX/J,I/):=TUX(/I+1,J/)'END', 'END', 'EOF',

```

Предназначена для определения оптимальных программ расходов воды из

водохранилищ каскада и изменений уровня воды у плотин (прогр. 52) [37]

**Используемые подпрограммы.** НОА, ОПТИМ

**Метод.** Главный задачей оперативного управления водохозяйственным комплексом является обеспечение желательных оптимальных уровневых режимов по всему бассейну для заданного интервала управления (месяц, сезон, год). При за-

```

79. 'PRINT' ,//(//,19НЕСОВЫЕ КОЭФФ.(MB1),/),,,,
80. 'PRINT' ,(F10.3),,MB1,, 'PRINT' ,(//,19НЕСОВЫЕ КОЭФФ.(MBZ),
81. /),,, 'PRINT' ,(F10.3),,MBZ,, 'FOR' I:=1 'STEP' 1 'UNTIL' Z1-1'DO'
82. XC1(/1/):=XC1(/1/),, 'FOR' I:=1 'STEP' 1 'UNTIL' L3'DO,IM(/I,1/)
83. :=2., 'FOR' K:=1 'STEP' 1 'UNTIL' L1'DO,'BEGIN' , 'FOR' I:=1 'STEP' 1
84. 'UNTIL' L3'DO,'BEGIN' X(/I/):=XC1(/(I-1)*V2+1/),, 'FOR' J:=2
85. 'STEP' 1 'UNTIL' L2'DO,X(/(J-1)*L4+1/):=X(/I/)'END',,
86. 'FOR' I:=1 'STEP' 1 'UNTIL' V2'DO,'BEGIN' K(/I+L3/):=XC1(/Z1-1-VB-
87. 1/),, 'FOR' I:=2 'STEP' 1 'UNTIL' L2'DO,X(/(J-1)*L4+1+L3/):=X(/I+
88. L3/)'END',, 'PRINT' ,//(13,18Н-й этап управления,/,21(Н-)),,
89. K.,OPTIM(X,IM,NO,ND,ML,LT,CQ,C,CS,TU,3,31,E,F,EQUAT,JACOBI,0T,L4),,
90. 'PRINT' ,//(10Х,ТИРЕЛЕНИЕ,/,9Х,10(Н-)),,, 'PRINT'
91. ,,(C,2H1F-,E20.9,/,2HХ=),,F,, 'PRINT' ,(10E12.5),,X,, 'FOR'
92. I:=1 'STEP' 1 'UNTIL' M-UT'DO,'FOR' J:=TAU,STEP,-1 'UNTIL' 1'DO,
93. XC1(/J+1+(I-1)*L2/):=XC1(/J+(I-1)*V2/),, 'FOR' I:=1 'STEP'
94. 1'UNTIL' M-15 'DO' XC1(/1+(I-1)*V2/):=TUX(/1,1/),, 'FOR' I:=1
95. 'STEP' 1 'UNTIL' L5'DO,LC1(/(L3+I-1)*V2+1/):=TUX(/1,1/),, 'FOR'
96. I:=1 'STEP' 1 'UNTIL' UV'DO XC1(/Z1-I-UV+1/):=TUX(/1,L6,I/),
97. 'FOR' I:=1 'STEP' 1 'UNTIL' L2+LL-1'DO,'FOR' J:=1 'STEP' 1 'UNTIL'
98. L5'DO IX/J,I/):=TUX(/I+1,J/)'END', 'END', 'EOF',

```

данной водохозяйственной обстановке, которая включает: прогноз приточности, начальные уровни и расходы для всех водохранилищ, а также требуемые уменьшаемые режимы, систему ограничений на режимные параметры, прогноз осадков, испарения и план водопотребления.

Для каскада, состоящего из  $N$  водохранилищ, в качестве меры отклонения уровневых режимов от заданных на интервале управления примем функционал

$$\phi = \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^M (h_k^i - h_k^{i*})^2 = \frac{1}{2} \sum_{i=1}^N \omega_i \phi^i, \quad (5.20)$$

где последовательность  $h_k^i$ ,  $k = \overline{1, M}$  является заданным (наилучшим) уровневым режимом  $i$ -го водохранилища,  $M$  — число тактов интервала управления, которое определяется длиной этого интервала (сезон, год) и временем дискретизации про-

цесса  $v (M = T/v)$ ;  $\phi_i = \sum_{k=1}^M (h_k^i - h_k^{i*})^2$  ( $i = \overline{1, N}$ ) — весовые коэффициенты водохранилищ, которые учитывают их важность при управлении речным стоком.

Задача оптимизации оперативного режима каскада водохранилищ (с применением векторных обозначений) формулируется следующим образом: определить оптимальную последовательность управляемых векторов (расходов воды в нижние бьефы плотин водохранилиши)  $q^0 k$  ( $k = \overline{1, M}$ ), минимизирующую функционал  $\Phi = \frac{1}{2} \omega^T \varphi$  при заданных прогнозах приточности  $P_k$ , обобщенных потерь  $\omega_k$  и при соблюдении ограничений в виде уравнения динамики каскада водохранилищ

$$\Psi_{k+1} = \Psi_k + v(Bq_{k+1} + \omega_{k+1}), \quad (5.21)$$

а также при выполнении ряда двусторонних ограничений-неравенств на уровни, сбросы и их приращения. Используются следующие обозначения:

$$\omega^T = \{\omega^i | i = \overline{1, N}\}; \quad (5.22)$$

$$\Psi_k^T = \{\Psi^i | h_k^i, q_k^i, i = \overline{1, N}\}, \quad k = \overline{1, M},$$

где  $\Psi^i (h_k^i, q_k^i, i = \overline{1, N})$ ,  $k = \overline{1, M}$ , — динамические характеристики водохранилища.

На практике прогнозы приточности и обобщенные потери уточняются на каждом такте управления, при этом могут также изменяться и требования к режиму каскада, что выражается в уточнении ограничений. Поэтому на каждом такте необходимо повторять решение задачи оптимизации с учетом новой водохозяйственной обстановки.

При указанных особенностях задачи применяют методы управления с оптимизацией на скользящем интервале [14] формально такое управление состоит в том, что каждый такт решается задача оптимизации на интервале постоянной длины  $L$ , но в системе реализуется лишь первое значение управления из всей последовательности, а в следующий такт расчет повторяется с сохранением длины интервала. Таким образом, интервал оптимизации «скользит», что позволяет эффективно учитывать в текущем управлении изменения, происшедшие в системе.

При скользящей оптимизации можно одновременно предопределить вычислительные трудности, связанные с оптимизацией системы большой размерности, и оперативно учесть изменения водохозяйственной обстановки (связанной, например, с уточнением прогнозов) при решении задачи оперативной оптимизации режима. При этом на каждом такте движения системы оптимизируется функционал

$$\Phi_L = \frac{1}{2} \omega^T \Psi_L, \quad (5.23)$$

где компоненты вектора  $\Psi_L$  имеют вид

$$\Psi_L^i = \sum_{k=n}^{k+L-1} (h_k^i - h_k^{i*})^2; \quad (5.24)$$

$n$  — индекс текущего такта, а оптимальное управление системой осуществляется управляющими воздействиями, определенными на каждом такте Условием

$$q_n^0 = q_k^0|_{k=n} \quad (5.25)$$

В условиях (5.25) последовательность  $q^0 k, k = \overline{n, n+L-1}$  является решением задачи оптимизации функционала (5.23) при заданном прототипе  $\omega_k$ , а также при выполнении ряда ограничений на уровнях, сбросы и их приращения в виде двусторонних неравенств:

$$h_k^t_{\min} \leq h_k^t \leq h_k^t_{\max}; \quad (5.26)$$

$$q_k^t_{\min} \leq q_k^t \leq q_k^t_{\max}; \quad (5.27)$$

$$(h_k^t - h_{k-1}^t)^2 \leq (\Delta h_k^t); \quad (5.28)$$

$$(q_k^t - q_{k-1}^t)^2 \leq (\Delta q_k^t). \quad (5.29)$$

Эту задачу сводим к эквивалентной задаче НЛП. Представим сначала ограничения (5.26) — (5.29) в виде односторонних (покомпонентно):

$$\begin{aligned} y_1^t(h_k^t, q_k^t) &= \Psi^t(h_{k+1}^t, q_{k+1}^t) - \Psi^t(h_k^t - q_k^t) - \\ &- v(q_{k+1}^{t-1} - q_{k+1}^t + \omega_{k+1}^t) = 0, \quad q_{k+1}^t = P_{k+1}^t; \end{aligned} \quad (5.30)$$

$$y_{2k}^t(h_k^t, q_k^t) = (h_k^t - h_k^t_{\min})(h_k^t - h_k^t_{\max}) \leq 0; \quad (5.31)$$

$$y_{3k}^t(h_k^t, q_k^t) = (q_k^t - q_k^t_{\min})(q_k^t - q_k^t_{\max}) \leq 0; \quad (5.32)$$

$$y_4^t(h_{k+1}^t, h_k^t) = (h_{k+1}^t - h_k^t)^2 - (\Delta h_k^t)^2 \leq 0; \quad (5.33)$$

$$y_5^t(q_{k+1}^t, q_k^t) = (q_{k+1}^t - q_k^t)^2 - (\Delta q_k^t)^2 \leq 0. \quad (5.34)$$

Введем общий вектор переменных

$$\bar{X}^T = \{h_k^t, q_k^t, i = \overline{1, N}, k = \overline{n, n+L-1}\} \quad (5.35)$$

размерностью  $m_0 = 2NL$ , вектор ограничений в виде равенств

$$y_p^T(\bar{X}) = \{y_{ik}^t, i = \overline{1, \bar{N}}, k = \overline{n, n+L-1}\} \quad (5.36)$$

размерностью  $l_0 = NL$ , вектор ограничений в виде неравенств

$$y_H^T(\bar{X}) = \{y_{ik}^t, i = \overline{1, \bar{N}}, k = \overline{n, n+L-1}, j = \overline{2, 5}\} \quad (5.37)$$

размерностью  $l_1 = LNL$  и функцию

$$f(\bar{X}) = \Phi_L = \frac{1}{2} \sum_{l=1}^N x^l \sum_{h=n}^{n+L-1} \gamma_k(h_{k+1}^t - h_{k+1}^{*t}). \quad (5.38)$$

Тогда для исходной задачи динамической оптимизации (5.21), (5.22), (5.26), (5.29) можно сформулировать эквивалентную ей (в силу выполненных преобразований) задачу нелинейного программирования: определить минимум квадратичной функции

$$f = f(\bar{X}) \quad (5.39)$$

от  $m_0$  переменных

$$x^T = \{x^1, x^2, \dots, x^{m_0}\} \quad (5.40)$$

при заданной системе  $l = l_0 + l_1$  ограничений в виде  $l_0$  равенств

$$y_p(\bar{X}) = 0 \quad (5.41)$$

и  $l_1$  неравенств

$$\bar{y}_H(\bar{X}) \leq 0. \quad (5.42)$$

Задача нелинейного программирования решается с помощью программы OptIMI.

**Описание параметров.** *Вводы/выходы:*

ПЧ — периодичность вывода на печать промежуточных результатов поиска

СQ — максимальное количество итераций;

G — коэффициент, определяющий поиск максимума ( $\gamma_0 > 0$ ) или минимума ( $\gamma_0 < 0$ );

СШ — точность выбора шага в задаче НЛП;

ТАУ — величина начального шага;

Е1 — точность решения ограничений;

Е2 — точность выполнения условия равенства плюс модуля градиента функ. или  $\sqrt{\epsilon}$ ;

Е3 — точность выполнения условия останова решения;

Н — число всех водохранилищ;

Т — длина периода оптимизации;

СТ — степень (максимальная) аппроксимирующих полиномов;

ДТ — длина интервала дискретизации;

М1 — наличие (любое число) или отсутствие (нуль) ограничений на приращения уровней и сбросов

NO — число основных переменных;

М1 — число ограничений на приращения уровней и сбросов;

ND — число дополняемых переменных;

М — число всех ограничений;

NK — число коэффициентов аппроксимирующего полинома.

Параметры массивов размерностью  $[1 : N, 1 : L]$ , относящиеся к скользящему интервалу:

НЖ — желаемые уровни верхних бьефов плотин водохранилищ;

НЛ, HG, QL, QG — минимально и максимально допустимые уровни и сбросы (Q)

погохранилищ соответственно;

DH, DQ — допустимые ограничения на приращения (скачки) уровней

(H) и сбросов;

P — обобщенные потери воды из водохранилищ;

КТ — весовые коэффициенты важности интервалов времени для водохранилищ.

Для полного интервала управления приняты следующие идентификаторы

Массы размерностью  $[1 : T, 1 : N]$ :  
НЖЕ — желаемые уровни верхних бьефов водохранилищ;

РБ — боковая пригожность;

ОС, ИС, ОТ — осадки, испарения и отъемы, соответственно;

ПОГ — обобщенные потери воды из водохранилищ;

КОТ — весовые коэффициенты важности интервалов времени для водохранилищ;

ХРАП — хранилиши (в данном варианте все интервалы времени равноправны — вес равен 1).

В программе используются также следующие параметры:

РОС [1 : 1] — прототип основной пригожности в верхнее водохранилище на полном интервале управления;

РО [1 : L] — основная пригожность на скользящем интервале оптимизации;

ХО [1 : NO] — основные переменные задачи нелинейного программирования, относящиеся к скользящему интервалу оптимизации;

ХН, ХГ [1 : N \* T] — оптимальные уровни воды в верхних бьефах плотин водохранилищ и оптимальные расходы в нижних бьефах плотин водохранилищ соответственно;

HQ [1 : N, 1 : 4] — постоянные ограничения на уровни и расходы для всех водохранилищ (минимальные и максимальные);

НО, ОИ, НИ, QI [1 : N] — начальные (и текущие) уровни и расходы;

W11, WQ, WI1 · N] — текущие значения производных по уровням и расходам,

а также объемы, а также коэффициенты (коэффициенты важности) водо-

хранилищ.

**Последовательность работы программы.** Структурная схема алгоритма, реали-  
зующего метод скользящей оптимизации при следении исходной динамической  
задачи к эквивалентной задаче линейного оптимизирования (НЛП), изображена  
на рис. 9.

Основными источниками данных для работы алгоритма являются следующие:  
число водохранилищ  $N$ , длина проксиро-  
ванного интервала управления  $M$ , длина  $L$ ,  
скользящего интервала оптимизации  $L$ ,  
матрица коэффициентов аппроксимации  
динамических характеристик водохрани-  
лищ (5.22) и исходная подоходзяйственная  
обстановка  $\Omega_0$ , которая включает уровни и  
бросы ( $h_0$  и  $q_0$ ) для всех водохранилищ,  
требуемые уровни — режимы  $h_k$ ,  $k =$   
 $= \overline{1, M}$ , систему ограничений на режимы  
ные параметры (5.21) — (5.29), а также  
прогнозы основной и боковой притонос-  
теи, осадков и испарений, планы водо-  
отведения.

Согласно алгоритму OPTIM1 на каж-  
дой итерации процедура поиска экстре-  
мума вычисляются функции (5.39), певяз-  
ки ограничений в виде равенств (5.41)  
и неравенств (5.42) (процедура EQUATI),  
а также те элементы вектора  $\bar{V}_k^t$  и матри-  
цы частных производных  $\bar{V}_{k,t}$ , которые  
зависят от основных переменных  $\bar{X}$  (про-  
цедура JACOBI). При вычислении певяз-  
зок ограничений и элементов матрицы  
частных производных определяются значение функции  $\Psi^i(h_k^t, q_k^t)$  с помощью под-  
программы WHQ и их частных производных по переменным  $\bar{X}$  с помощью подпро-  
граммы DWHQ.

В результате формирования и решения задачи НЛП по алгоритму QOPTIM1  
получаем на каждом  $n$ -м такте движения системы вектор  $\bar{X}^n$ ,  $h_n^{opt}$ ,  $q_n^{opt}$ ,  $k = n$ ,  
 $\overline{n+L-1}$ . Значение вектора управления  $q_k^{opt}$ , согласно принципу управления  
с оптимизацией на скользящем интервале, реализуется на объекте (при работе  
в реальном времени) или запоминается. Затем сдвигается вперед интервал оптими-  
зации длины  $L$ , и формируется задача НЛП для интервала  $[M+1, n+L]$ ; при  
этом водохранилища обстановка корректируется. Через  $M - L$  тактов алгоритм  
останавливается и выдает решение в виде оптимальных программ управляющего  
 $q_n^{opt}$  и регулируемого  $h_n^{opt}$  векторов  $n = 1, M$ , куда полностью входит решение  
последней задачи оптимизации на интервале  $[M - L + 1, M]$ . Кроме того, выво-  
дится оптимальное (результирующее) значение показателя качества:

$$\phi^{opt} = \frac{1}{2} \sum_{i=1}^N \lambda^i \sum_{n=1}^M \mu_n (h_n^{l,con} - h_n^{*})^2. \quad (5.43)$$

Эта величина, вычисленная при различных  $L$  с помощью тестовых расчетов, слу-  
жит для установления факта «насыщения» и стабилизации параметров оптимальной  
системы.

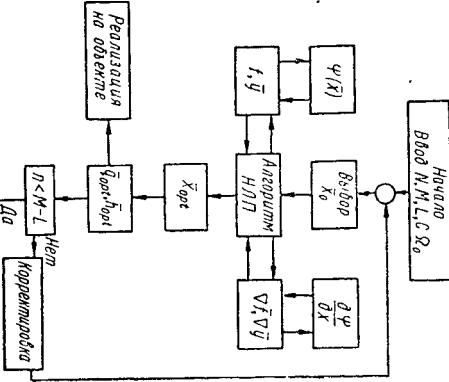


Рис. 9. Структурная схема алго-  
ритма оптимизации режима каска-  
да водохранилищ

Предложенная программа обладает достаточно широкими возможностями. На-  
пример, корректируя систему ограничений, а также требуемые уровни режимы  
водохранилищ, можно пойти для той или иной водозапасной ситуации ре-  
шения, удовлетворяющее в лосасточной степени всех участников водохозяйствен-  
ного комплекса. Кроме того, после минимальных изменений с помощью рассмотрен-  
ной программы можно оптимизировать режим каскада водохранилищ по любо-  
му другому заданному критерию.

Для работы с программой необходимо ввести последовательно значения вход-  
ных параметров и массивов HQ, НЖЕ, РОС, РВ, ОС, ИС, ОГ, КА, КВ, ИО, КО.

#### ПРИГРАММА 52

```

1. 'ALGOL':OPTIM1,
2. 'BEGIN' I:INTEGER;N,T,CT,PI,C0,L,M,M1,I0,N,D,NK,I,J,H,, 'REAL',DT,
3. C,CL,TAV,E1,E2,E3,F,,READ(CP,CO,G,CM,TAV,E1,E2,E3,N,T,CT,DT,
4. L,M1)..'PRINT',//,11,6)3)A100/,10A,9(1H)//,
5. 2X,ЭНЧ=,13,34,ЭНС#=,13,3X,2ИГ=,E10,3,3X,ЭНГ#=,E10,3,/2Х,
6. 4ИТАУ=,E12,4,3Х,ЭНБ=,E12,4,3Х,/ЭНЗ=,E12,4,//,
7. 2X2ИИ=13,342ИТ=13,3ХЭНС#=13//,2XЭНГ#=12,4,5Х13ИУПРЕДЕЛЕНИЕ ,
8. 2ИИ=,13)',ИЧ,СУ0'G,0,СУ1'AV,E1,E2,E3,M,T,GT,L,,IF'M1=0
9. 'THEN','PRINT',//3ИИ=0 - ЕЩЕ ОГРАНИЧЕНИЯ НА DH,DO)','ELSE'
10. 'BEGIN' M:=2*I*L,,'PRINT',//5Х2ИНЕСТЬ ОГРАНИЧЕНИЯ НА DH,DO),
11. 'END',M:=2*I*L,,M:=2*I*L+1,M:=2*N*L+1,,MK=(CT+1)*(CT+2)
12. /2-1..,'PRINT',/(5ХЭНД=13,ЭХЭНЮ=13,ЭХЭНД=13,Э2ИИ=13),
13. NO,NД,M,, 'BEGIN' AND AY'10,HL,HG,OL,OG,DM,DO,IT,P/(1:N,1:L),
14. P0/(1:I:L),IA(1:N,1:M)/,B,IC,II,0,0,WI,WI,WI/(1:N),НЕ,РЕ,ОС,
15. ИС,(T,ИС,КОТ,КОТ/(1:T,1:N/),НОC/(1:T/),НО/(1:N,1:4/),ХО/(1:N/),
16. ХИ,ЛО/(1:H*T/),,'PROCEDURE'WHQ(JB,H,O,W),,'INTEGER'NB,, 'REAL',
17. H,O,W,, 'BEGIN' INTEGER'I,J,K,,W:=KA(O,NB/),I=0,, 'FOR',J:=1
18. 'STEP'1'UNTIL'CT'DO,'FOR',K:=0'STEP'1'UNTIL'J'DO,'BEGIN'I:=I+1
19. .W:=W+KA(I,NB/)*I**(J-K)*Q**K'END','END'WHQ,, 'PROCEDURE'
20. ДМОКUB,I,O,W,WU,, 'INTEGER'NB,, 'REAL'H,O,WH,WU,, 'BEGIN'
21. 'INTEGER',I,J,K,,I:=0,,WH:=W:=0,, 'FOR',J:=1'STEP'1'UNTIL'CT'DO,
22. 'FOR',K:=0'STEP'1'UNTIL'J'DO,'BEGIN'I:=I+1,, 'IF'K'NE'J'THEN'
23. WH:=WH+KA(I,NB/)*K**((J-K)*(K-1))**K,, 'IF'K'NE'0'THEN'WQ:=
24. WQ+WA(I,NB/)*K**((J-K)*(K-1))**K'END,'END',DWNQ,, 'PROCEDURE'
25. EQUAT(I,F,Y,,,'ARRAY'X,Y,,,'REAL'F,, 'BEGIN' INTEGER'I,J,H,,'
26. 'REAL'R,V,, 'ARRAY'P1,PC(1:N/),,'FOR',J:=1'STEP'1'UNTIL'N'DO'
27. 'UNTIL'L'DO,'BEGIN'V:=P0(V/Y,,,'FOR',J:=1'STEP'1'UNTIL'N'DO'
28. 'UNTIL'L'DO,'BEGIN'V:=P0(V/Y,,,'FOR',J:=1'STEP'1'UNTIL'N'DO'
```

```

29. BEGIN' H:=I+1., F:=F+IF(B/J)*KT(J,I/)*(X(I/V)-HE(J,I/))*?*, .
30. Y(V/J):=(HL(J,I/)-Y(V/J/))*(HG(J,V/)-X(H/)).., X(H/):=H/V/,
31. (OL(J,I/)-X(VN*L+V/)*LG(J,I/)-X(H*L+V/)), R:=-WT(J/),
32. WHQ(J,X(I/V),X(H*L+V/),WI(J/)), R:=R*WH(J/)-Y*(V-
33. X(N*L+V/)*P(J,I/)), V:=X(H*L+V/), END, END, , IF'M'NE'0
34. , THEN, BEGIN, FOR J:=1'STEP'1'UNTIL'N'DO, BEGIN'PRH(J/):=
35. H0(J/), , RO(J/):=W0C(J/), END, , H:=0., , FOR I:=1'STEP'1
36. 'UNTIL'1'DO, 'FOR J:=1'STEP'1'UNTIL'N'DO, WET'I'H:=H+1.,
37. R:=X(V/J)-RH(J/), V:=X(H*L+V/)-RO(J/), , Y(N0+V/):=I*2
38. -DH(J,I/)*2., Y(H-M1+V/):=I*2-DO(V,J,I/)*2., RH(J/):=
39. X(H/), , RO(J/):=X(H*L+V/)'END, END, END'EQUAT,
40. 'PROGRAM'EJACOBIX(X,NF,NY), 'ARRAY'X,NF,NY,, 'BEGIN'
41. , INTEGER'I,J,H,, 'REAL'R,V,, 'ARRAY'R,H,RO(J,I/V).., H:=0.,
42. , FOR I:=1'STEP'1'UNTIL'N'DO, 'FOR J:=1'STEP'1'UNTIL'N'DO,
43. , BEGIN'H:=H+1., NF(V/H):=IK(J,I/)*KT(J,I/)*(X(I/V)-
44. HL(J,I/)), , DWHQ(J,X(I/V),X(H*L+V/),WH(V/J/),WC(J/)), ,
45. NY(V/H,V/):=2*X(V/J)-HL(J,I/)-HG(J,I/), ,
46. NY(N L+V/H):=2*X(H*L+V/)-OL(J,I/)+DT., 'IF',
47. NY(N H,H,V/):=WH(J/), NY(H*A+H,N*B+H/):=WO(J,I/), ,
48. H'L*H*(I-1), THEN, BEGIN'NY((N*D+N*H,H/):=WH(V/), ,
49. NY((N*T+N*H,N*L+V/):=WQ(J,I/)) END, END, , H:=0., , FOR I:=1
50. 'STEP'1'UNTIL'1'DO, BEGIN'HE:=H+1., 'FOR J:=2'STEP'1'UNTIL'N'DO,
51. N'DO, 'BEGIN' H:=H+1., NY(N*D+H,N*L+H-1/):=-DT, END, ,
52. , IF'M'NE'0'THEN, BEGIN, 'FOR J:=1'STEP'1'UNTIL'N'DO'
53. , BEGIN'RH(J/):=H0(J/), RO(J/):=W0(J/), END, , H:=0.,
54. , FOR T:=1'STEP'1'UNTIL'1'DO, 'FOR J:=1'STEP'1'UNTIL'N'DO,
55. , BEGIN'JL:=H+1., R:=X(H/)-RH(J/), , V:=X(H*L+V/)-RO(J/), ,
56. NY(N0+H,H/):=2*R, NY(H-M1+H,N*L+H/):=2*V
57. , THEN, BEGIN'NY((N0+H,H-H/):=-2*R, NY(H-M1+H,N*L+H-1/):=2*V
58. , END, , RH(J/):=X(H/), , RO(J/):=X(H*L+V/), END, END,
59. , END JACBI, READIO,HAT,POC,PB,OC,IC,OT,RA,KB,H0,00), ,
60. , PRINT'', (2X3HKB/(6E12.4)), , HC, , PRINT'', (2X4HPCG=/, ,
61. PRINT'', (2X4HPCG=/(10E12.4)), , HME, , PRINT'', (2X4HPCG=/, ,
62. (10E12.4)), POC, , PRINT'', (2X3HFB/(10E12.4)), , PE, , PRINT',
63. , (2X3HFB/(10E12.4)), , OC, , PRINT'', (2X3HFB/(10E12.4)), ,
64. , IC, , PRINT'', (2X3HFB/(10E12.4)), , OT, , PRINT'', (2X3HFB=/
65. (10E12.4)), , KB, , PRINT'', (2X21HQB03Ф.ЛНПОКСИМ,Б)ХР=H/V/,
66. (5X10E12.5)), , KA, , PRINT'', (2X3H00=/(4X6F9.2)), , H0,
57. , PRINT'', (2X3H00=/(4X6F9.3)), , 00,
68. , FOR I:=1'STEP'1'UNTIL'T'DO, 'FOR J:=1'STEP'1'UNTIL'N'DO'
69. , BEGIN'NOT(V,I,J/):=PB(V,I,J/)+OC(V,I,J/)-WC(V,I,J/), ,
70. NOT(V,I,J/):=1.0'END, , H:=0., , FOR I:=1'STEP'1'UNTIL'L'DO,
71. , FOR J:=1'STEP'1'UNTIL'N'DO, BEGIN'H:=H+1., HL(J,I/):=
72. HL(J,I/), , LG(J,I,I/):=LO(J,I,2/), , LM(J,I/):=HME(V,I,J/),
73. OL(J,I/):=HO(J,I,3/), , OG(J,I,I/):=HO(J,I,4/), , P(V,I,I/):=
74. NOT(V,I,J/), , OR(V,I,I/):=NOT(V,I,J/), , X0(V/I):=HE(V,J,I/),
75. X0(VN*L+V/):=PO(V/I)'END, , H:=0., , ME:H:=H+1., 'IF'H'NE'0'THEN',
76. , PRINT'', (/7X11HGLOBAL STEPT2/6X1(11-))'', H., OPTM(X0,M,NO,
77. H4,H4,CO,G,CJL,TAY,E1,E2,E3,F,EQUAT,I,JACBI), , IF'M'NE'0
78. , THEN, BEGIN'PRINT'', (/7X14HRESULT OH STEPT3)'', H., PRIN'
79. , '(8X2IF=E1.5/8X2HX/(7X10E12.5))'', F, X0'END, , IF'H=T-L+1
80. , THEN, 'GOTO'FTH, , 'FOR I:=1'STEP'1'UNTIL'L-1'DO, BEGIN'P0(/V)
81. :=H0(I+1/), , 'FOR J:=1'STEP'1'UNTIL'N'DO, BEGIN'P(V,I,I/):=
82. P(V,I,I+1/), , KT(V,I,I/):=KT(V,I,I+1/), , HK(V,I,I/):=HE(V,I,I+1/),
83. , END, , END', , PO(V,I,I/):=PO(L+H/), , 'FOR J:=1'STEP'1'UNTIL'N
84. , DO, , BEGIN'UR(V,I,I/):=HE(L+H,J/), , P(V,I,I/):=NOT(L+H,J/), ,
85. KT(V,I,I/):=KOT(L+H,J/), , HO(V,I,I/):=X0(V,I,I/), , Q0(V,I,I/):=
86. X0(H*L+J/), , XH((H-1)*N+V/):=X0(V,I,I/), , X0((H-1)*N+V/):=
87. X0(H*L+J/)*1000'END, , 'IF'H'LE'T-L'THEN' GOTO'ME, FTH:H:=0.,
88. , 'FOR I:=1'STEP'1'UNTIL'L'DO, , 'FOR J:=1'STEP'1'UNTIL'N'DO'
89. , BMGTN'U:=H+1., XH((T-L)*N+V/):=X0(H/), , XQ((T-L)*N+V/):=
90. X0(H*L+H/)*1000'DID, , 'PRINT'', //2X18HOUT.YP0BH (METP)/,
91. X20(1H-)/(20X2P9.2)), , XH, , 'PRINT'', //2X12HOUT.CBPOCHI,
92. 14H(RYE,NETP/CER)/X28(1H-)/(20X2P9.0)), XQ, , 'IF'H'NE'0
93. , 'THEN' , BEGIN'P:=0.0., H:=0., 'FOR I:=1'STEP'1'UNTIL'T'DO, 'FOR'
94. J:=1'STEP'1'UNTIL'N'DO, 'BEGIN'H:=H+1., P:=P+KB(V,J/)*(XH(V)
95. -HE(V,I,J/))*2'END, , 'PRINT'', //2X19HOPTIMAL FUNCTION P=,
96. E12.5), , 'END'END'END'EOP,

```

## Глава 6

# ВЫДЕЛЕНИЕ МНОЖЕСТВА ПРИЗНАКОВ И РАСПОЗНАВАНИЕ ОБРАЗОВ

## 1. ПРИМЕНЕНИЕ ПРИНЦИПОВ САМООРГАНИЗАЦИИ ДЛЯ ОЦЕНКИ МИНИМАКСНОГО КРИТЕРИЯ КАЧЕСТВА АНСАМБЛЯ ПРИЗНАКОВ

В настоящее время задачи распознавания решаются в многомерном пространстве признаков. Для получения более компактных распознающих систем исходные непрерывные признаки обычно бинаризуются и, кроме того, для уменьшения степени влияния помех, усредняются в пределах некоторого интервала времени. Следовательно, на этапе обучения распознающей системы необходимо решать следующие три задачи: выбор порогов для нелинейных преобразователей, отбор ансамбля наиболее информативных признаков и выработка решающего правила. Обычно эти задачи решаются последовательно, а информативные признаки отбираются по индивидуальному критерию.

Однако выбор ансамбля признаков по количественному критерию, когда оценивается полезность того или иного ансамбля признаков в целом, дает значительно лучшие результаты в смысле точности распознавания. При большом числе признаков решение задачи оценки ансамбля признаков перебором невозможно из-за большого времени счета и требующихся при этом больших объемов оперативной памяти.

Применение принципа самоорганизации позволяет значительно сократить время счета и требующийся объем оперативной памяти. При распознавании образов по алгоритмам самоорганизации все три указанные задачи решаются параллельно: в конце обучения алгоритм распознавания выдает оптимальные в смысле вероятности значения порогов для каждого признака, ансамбль наиболее информативных признаков и решает правилу.

**Выбор оптимальных значений порогов.** Во входном устройстве распознающей системы непрерывные признаки при помощи простейших нелинейных преобразований (порогов) переводятся в бинарные или двоичные коды. Оптимальные значения порогов выбираются либо определением распределений вероятности, либо перебором вариантов их дискретных значений. В первом случае этап построения кривых распределения вероятностей предшествует переводу непрерывных признаков в бинарные.

Если датчики содержат только один нелинейный пороговый элемент, то оптимальным значением порога является такое, при котором изображения одного класса получают знак «плюс», а другого — «минус».

Однако в современных датчиках признаков применяются более сложные нелинейные преобразования, использующие несколько порогов. Например, первый порог разделяет входные величины по амплитуде, второй — по так называемому «каполнению пачки импульсов» и т. п. Применяются датчики с двумя, тремя и большим числом пороговых элементов. При этом вопрос о выборе оптимальных значений порогов не так прост, как при одном пороге.

Вероятностные методы определения оптимальных порогов не применимы в этом случае, так как построение многомерных вероятностных распределений по экспериментальным данным практически невозможно: требуется слишком много экспериментальных данных и длительные вычисления.

Вопрос можно решить перебором вариантов, что также не просто из-за большого числа перебираемых вариантов. При двух порогах, каждому из которых можно залатать, например, десять дискретных значений, для каждого признака нужно перебрать сто вариантов.

Алгоритмы самоорганизации позволяют не только выбирать оптимальный (по заданному критерию) ансамбль признаков, но и найти соответствующее ему соглашение порогов при помощи критерия числа разрешаемых споров, как для каждого отдельного признака, так и для ансамбля из двух и более признаков.

Отдельный признак (или ансамбль признаков) оценивается на информационном уровне, т. е. без учета качества распознающей системы.

Согласно индивидуальному критерию [21] признаки каждой пары образов сравниваются между собой. Спор считается решенным, если данный бинарный признак имеет различные знаки для двух изображаемых образов, и перешелним, — если знаки одинаковы. Перебрав попарно (в случае полного перебора) все изображения, составляем карты разрешаемых споров для каждого признака; разрешенный отмечается единицей (1), нерешенный — нулем (0). Кarta представляет собой квадратную матрицу из единиц и нулей. Строки соответствуют одному из поларно сравниваемых образов, а столбцы — другому. Число карт равно числу комбинаций образов по два.

Составив карты для всех заданных признаков, можно оценить каждый из них индивидуально. Тот признак лучше, у которого карта содержит больше единиц (разрешаемых споров). Составление карт повторяется для всех значений, что дает возможность выбрать их оптимальные значения: наилучшие значения порогов соответствуют карте, содержащей максимум единиц. Если ряд соседних (по величине порогов) карт содержит равное число единиц, то в качестве оптимальной выбирается средняя карта. Так решается вопрос о выборе наибольшего количества признаков и значений порогов по индивидуальному критерию числа разрешаемых споров.

Суммируя данные карты двух отдельных признаков, получаем суммарные карты, оценивающие простейший ансамбль состоящий из двух признаков. Суммарные карты могут содержать двойки, единицы и нули. Нуль означает, что данная пара признаков ни одно спор не решает. Если суммарная карта не имеет нулей, можно построить распознающую систему, работающую только по двум признакам. Оптимальный выбор наиболее эффективной пары признаков и оптимального значения порогов основан на так называемом анализе минимумов. На всех картах выбираются клетки с минимальной суммой разрешаемых споров; оптимальной считается карта такой пары признаков, у которой эта минимальная оценка наибольшая (минимаксный критерий). Число разрешаемых споров, выбираемое по минимаксному критерию, показывает минимальную кратность решения споров для данной пары признаков.

Суммируя карты для трех, четырех и более признаков, получаем соответствующие суммарные карты, и для каждого ансамбля снова находим минимаксный критерий чистая разрешаемых споров.

Размер ансамбля все время возрастает до тех пор, пока не будет получена требуемая по техническим условиям кратность числа разрешаемых споров, или до окончания полного перебора всех возможных размеров ансамблей. Если минимальная кратность решения споров перестала увеличиваться, то дальнейшее усложение ансамбля бесполезообразно.

Каждая карта решаемых споров определяет собой эффективность некоторого признака или ансамбля признаков. Необходимо найти такую карту (один ансамбль), в которой минимальное число решаемых споров равно или больше заданной величине  $S_{\min}$ .

Другими словами: в программе поиска задается некоторое минимально допустимое число решаемых споров  $S_{\min} \geq 1$ , и все цифры искомой таблицы должны быть равны или, больше этого числа. Число  $S_{\min}$  называется показателем запаса разрешающей способности распознающей системы. Основной критерий склонен иметь минимаксный характер. Из всех карт выбирается та, в которой минимальное число решаемых споров максимальное и больше некоторой заданной величины  $S_{\min}$ , показывающей запас разрешающей способности системы.

Если пайтение число мало, то число признаков нужно увеличить, а их состав изменить.

Если отобрано несколько карт с одинаковыми наименьшими чистыми, следует воспользоваться дополнительными (вспомогательными) критериями.

Часто случается, что основному критерию удовлетворяет несколько карт. Тогда используются вспомогательные критерии селекции, применяемые последовательно один за другим:

**Критерий минимума числа признаков, входящих в ансамбль.** Из равных по

основному критерию карт выбираются те, которые отвечают меньшему числу признаков. Например, пусть задано  $S_{\min} \geq 1$ . Тем самым все карты, в которых имеются нули (непрещенные споры), отбрасываются. Остались, предположим, 20 карт, из которых пять нулей. Из них двум признаком отвечают 2 карты, трем — восемь карт, оставшиеся отвечают большому числу признаков. Тогда «победителями» селекции являются две карты с двумя признаками.

*Критерий суммы решаемых споров.* Преимущество получает карта, в которой сумма решаемых споров больше.

Таким образом, соблюдается следующая иерархия критериев:

- первый (основной) — минимаксный критерий чиста решаемых споров;
- второй — минимум числа признаков (критерий просчета ансамбля);
- третий — критерий суммы решаемых споров во всей карте.

#### Исходные значения непрерывных признаков

N <sub>то-</sub> ки	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	N <sub>то-</sub> ки	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	N <sub>то-</sub> ки	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>	N <sub>то-</sub> ки	V <sub>1</sub>	V <sub>2</sub>	V <sub>3</sub>
1	-1	1	-10	11	-1	7	-1	21	6	4	7	31	8	3	8
2	3	5	-7	12	7	2	2	22	6	6	32	3	8	0	0
3	0	-8	13	3	4	3	23	15	1	10	33	-2	9	7	-4
4	-5	12	-11	14	3	3	10	24	11	3	5	34	3	6	-4
5	2	11	0	15	0	2	7	25	8	2	3	35	7	7	3
6	-2	8	1	16	-3	8	6	6	10	5	0	36	3	3	-2
7	5	3	-5	17	-2	6	-2	27	5	9	-5	37	10	1	7
8	-4	1	3	18	1	4	1	28	8	9	38	6	4	-3	6
9	9	0	-7	19	8	1	4	29	4	10	-10	39	8	5	-2
10	1	6	2	20	4	4	9	30	4	6	-7	40	10	8	1

Соблюдение получения критериев особенно важно при многорядных селекционных алгоритмах, осуществляющих принцип самоорганизации.

Рассмотренный выше минимаксный критерий следует применять только в случае зависимых признаков. Для независимых признаков достаточно использовать различные индивидуальные критерии.

Свойство зависимости признаков поясним при помощи карт решаемых споров. Допустим левая часть карты первого признака заполнена нулями, а правая — единицами. Наоборот, для второго признака левая часть заполнена единицами, а правая — нулями. Такие два признака по индивидуальному критерию не годятся для распознавания. Однако вместе они полностью решают задачу, так как нули одной карты покрываются единицами второй. Только коллективный критерий может обнаружить такую пару зависимых признаков. Применение индивидуального отбора для зависимых признаков неэффективно.

**Пример 6.** Выбор ансамбля признаков. Изменение трех признаков  $V_1$ ,  $V_2$  и  $V_3$  на протяжении 40 тактов представлено в табл. 8.

Первой операцией обработки данных является операция бинаризации сигналов по амплитуде. Начальное значение порога по амплитуде  $\Theta_1$  (которое в дальнейшем будет варьироваться) выбираем так, чтобы одна половина сигналов была больше порога, а вторая — меньше его. Основанием для такого выбора является известная теорема Видроу, согласно которой оптимальные значения бинарного порога находятся вблизи значений, при котором в половине случаев сигнал превосходит порог, а в другой — меньше его. Пусть начальные значения порога по амплитуде равны  $\Theta_{11} = 3$ ,  $\Theta_{12} = 4$ ,  $\Theta_{13} = 0$  (первый индекс указывает номер порога, второй — номер признака), и давай следующий результат бинаризации:

Таблица 8

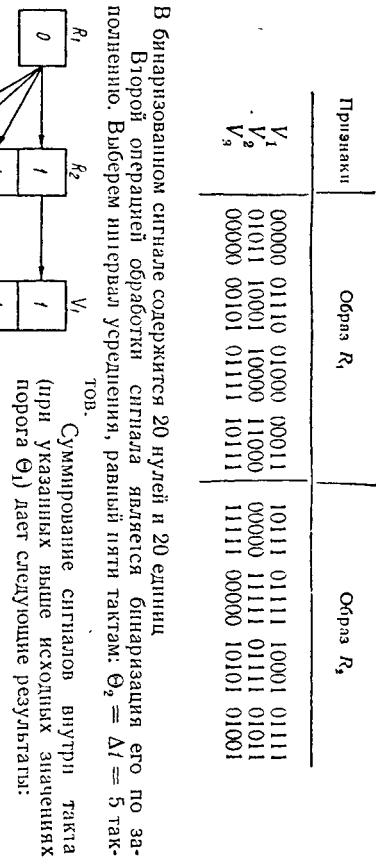


Рис. 10. Схема получения первой строки карты (признак  $V_1$ )

Выберем начальное значение порогов по заполнению так, чтобы в половине случаев получить 1, а в другой половине — 0. При  $\Theta_{21} = 2$ ,  $\Theta_{22} = 2$ ,  $\Theta_{23} = 2$  получим следующие результаты:

Признаки	Образ $R_1$	Образ $R_2$
$V_1$	0 1 0 0	1 1 0 1
$V_2$	1 0 0 0	0 1 1 1
$V_3$	0 0 1 1	1 0 1 0

Таким образом, непрерывные признаки, представленные в табл. 8, при помощи нелинейного преобразования с двумя порогами (по амплитуде и по заполнению) преобразованы в бинарный код.

Рассмотрим построение карт решаемых споров при вышеупомянутых (начальных) значениях порогов. Каждый элемент столбца для первого образа первого признака сравнивается со всеми элементами столбца второго образа этого же признака. Если элементы совпадают по значению, то в таблицу решаемых споров заносится 0, если не совпадают — 1.

Построение первой строки карты проиллюстрировано на рис. 10.

Карты решаемых споров для одиночных признаков имеют вид:

Признак $V_1$	Признак $V_2$	Признак $V_3$
1 1 0	1 0 0	1 0 1
0 0 1	0 1 1	0 1 0
1 1 0	0 1 1	0 1 0
1 1 0	0 1 1	0 1 0

По индивидуальному критерию признаки  $V_1$  и  $V_2$  оказались лучшими, так как в картах решаемых споров оказалась месть нулев (перешенных споров). Карты ансамблей по два и три признака получаются при суммировании соответствующих индивидуальных карт. При указанных значениях порогов получим карты ансамблей из двух признаков

Ансамбль $V_1 V_2$	Ансамбль $V_1 V_3$	Ансамбль $V_2 V_3$
2 1 0 1	2 1 1 1	2 1 0 1
0 1 2 1	1 0 2 0	1 0 2 0
1 2 1 2	1 2 0 1	1 2 0 1
1 2 1 2	1 2 0 2	1 2 0 1

и из трех признаков  $V_1 V_2 V_3$ :

3 1 1 1		
1 1 3 1		
1 3 1 3		
1 3 1 3		

Из рассмотрения карт следует, что признаки, взятые по одному, не решают задачу распознавания; в матрицах имеются нули (перешепные споры). Задачу распознавания решает ансамбль из двух ( $V_1 V_2$ ) признаков и ансамбль из трех признаков. Таким образом, признак  $V_3$  не нужен, так как его применение не увеличивает минимальное число решаемых споров (равно единице). Рассмотренное построение карт решаемых споров повторяется для всех комбинаций значений порогов  $\Theta_1$ ,  $\Theta_2$  и  $\Theta_3$ , начальные значения которых указаны выше.

Данные расчета:

число классов  $a = 2$ , число признаков  $b = 12$ ; число изображений в каждом классе  $c = 16$ ; число комбинаций значений порогов  $d = 100$

Исходя из этих данных, можно подсчитать: число карт  $\sum_{i=1}^b c_i d^i = \sum_{i=1}^{12} c_{12} 100^i =$

$= 1 \cdot 10^{26}$ , число клеток в карте  $c^2 = 256$ , число клеток всех карт, которые необходимо заполнить цифрами,  $S = c^2 \cdot \sum_{i=1}^b c_i d^i = 16^2 \sum_{i=1}^{12} c_{12} \cdot 100^i \approx 3 \cdot 27^{22}$ .

Объем полного детерминированного перебора настолько велик, что возникает задача его декомпозиции (уменьшения). Такое уменьшение достигается применением принципов самоорганизации.

**Процедура самоорганизации (селекции).** *Первый ряд селекции* В первом ряду отбираем (по индивидуальному критерию) некоторое заданное число карт, имеющих наибольшее число единичных. Например, если число заданный равно 100, то селектор выбирает по  $F = 10$  лучших карт для каждого признака. Всего будет отобрано  $N = Fb$  карт (где  $F$  — целое число, большее единицы, оценивающее свободу выбора) Д. Гавора.

*Второй ряд селекции.* Для всех карт, отобранных в первом ряду, строятся суммарные карты разрешаемых споров для ансамблей, содержащих два признака, всего  $F^2 b$  ( $b = 1$ ) карт. Из них селектор выбирает споры только  $N$  лучших (по минимаксному критерию) карт.

*Третий ряд селекции.* Для всех карт, отобранных во втором ряду, строятся суммарные карты числа разрешаемых споров для ансамблей, содержащих три признака. Всего будет составлено  $F^2 b$  ( $b = 2$ ) карт. Из них споры отбираются только  $N$  лучших (по минимаксному критерию) карт.

Аналогичным образом строятся четвертый, пятый и последующие ряды селекции. В каждом ряду можно составить  $F^2 b$  ( $b = S + 1$ ) различных карт для ансамблей из  $S$  признаков. Во всех рядах, кроме первого, действует минимаксный критерий отбора.

**Правило остановки.** Нарядуование рядов селекции продолжается до тех пор, пока увеличивается значение минимаксного критерия или пока не будет получена последняя карта ансамбля, содержащего все признаки. В последнем ряду селекции выбирается одна единственная карта, соответствующая лучшему ансамблю. При обратном просмотре ее построения получаем отрез об оптимальной комбинации значений порогов для каждого порождающего выходную карту признака.

Для рассмотренного выше примера при  $N = 3$ ,  $N_1 = 40$ ,  $F = 3$ ,  $\Theta_3 = 5$ .

Таблица 4

Пороги	Признаки		
	$V_1$	$V_2$	$V_3$
$\Theta_1^1$	9,9055	4,2124	-0,4822
$\Theta_2^1$	0	0,4	0,6
$\Theta_1^2$	-0,3767	5,0084	1,0030
$\Theta_2^2$	0,8	0,2	0,4
$\Theta_1^3$	0,9086	5,0084	-6,4233
$\Theta_2^3$	0,8	0,4	0,8

По минимаксному критерию отбираются ансамбли признаков  $V_1$ ,  $V_2$  при оптимальных значениях порогов  $\Theta_{11}^1 = 9,9055$ ,  $\Theta_{21}^1 = 0$ ,  $\Theta_{12}^1 = 4,2124$ ,  $\Theta_{22}^1 = 0,4$ , для которого минимальная кратность разрешаемых споров равна 2 (табл. 9). Кarta решаемых споров для этого ансамбля и указанных порогов имеет следующий вид.

2	2	2	2
2	2	2	2
2	2	2	2

**Расчет объема вычислений при самоорганизации.** Для тех же данных, которые были использованы в расчете объема вычислений при полном переборе, получим следующий объем вычислений при самоорганизации:

$$S = bdc^2 + \frac{1}{2} F^2 b^2 (b - 1)c^2 = 12 \cdot 100 \cdot 256 + \frac{1}{2} \cdot 10^2 \cdot 12^2 (12 - 1) 256 = 307200 + 20275200 = 20582400$$
 клеток карт, что вполне приемлемо для современных вычислительных машин. Таким образом, уменьшение объема перебора достигнуто.

## 2. ПРОГРАММА ВЫБОРА АНСАМБЛЯ ИНФОРМАТИВНЫХ ПРИЗНАКОВ НА ОСНОВЕ ПРИЧИПА САМООРГАНИЗАЦИИ

Предназначается для выбора ансамбля информативных признаков и оптимальных значений порогов нелинейных преобразователей распознавающей системы

(прогр. 53) [30].

**Язык программирования АЛГОЛ, ЭВМ БЭСМ-6.**

**Использование подпрограммы.** Нет.

**Метод.** Пусть имеется  $N$  признаков, каждый из которых измерен в  $N_1$  точках. Первые  $N_1/2$  точек относятся к первому образу, в следующие  $N_1/2$  точек — ко второму (рассматривается задача разделения на два класса). Необходимо для каждого признака найти наилучшее соотношение трех порогов:

по амплитуде  $\Theta_1$ ; по коэффициентам заполнения пачки  $\Theta_2$  (отношение числа единичных значений признака внутри интервала усреднения к длине интервала  $\Delta t$ ); длине интервала усреднения  $\Delta t$ .

Оптимальное значение порога  $\Theta_{1i}$  ищем внутри эффективного интервала изменения входной величины, в качестве верхней и нижней границ которого принимаются соответственно такие величины, чтобы выше и ниже их было расположено 90% всех имеющихся точек обоих классов.

В соответствии с принципами самоорганизации в первом ряду селекции для каждого признака по индивидуальному критерию отбираем заданное число  $F$  комбинаций порогов  $\Theta_{1i}$  и  $\Theta_{2i}$  для фиксированной длины интервала усреднения  $\Delta t$ . Таким образом, после первого ряда селекции будет отобрано  $NF$  карт разрешаемых споров. Бинаризация производится по следующему правилу: значение признака равно единице, если внутри интервала усреднения число точек, для которых  $\Theta_{1i} > 0$  больше, чем  $\Theta_{1i}\Delta t$  ( $0 \leq \Theta_{1i} \leq 1$ ). На втором ряду [21] покадочным сложением суммируемых карт решаемых споров составляем суммарные карты ( $F_2(N - 1)$  карт) для возможных ансамблей из двух признаков. Лучшими считаются карты, содержащие минимальное число клеток с максимальным значением минимальной суммы (минимаксный критерий). По окончании второго ряда будет отобрано  $V$  лучших суммарных карт.

Третий, четвертый и последующие ряды строятся аналогичным образом. В каждом ряду можно составить  $F_2(N - 3 + 1)N$  различных суммарных карт разрешаемых споров из  $S$  признаков. В последнем  $N_m$ -м ряду селекции будет составлено  $NF^2$  ансамблей, состоящих из  $N$  признаков. После каждого ряда селекции из всех отобранных ансамблей признаков выбирается наилучший в смысле минимаксного критерия.

Таким образом, после выполнения  $N$  рядов из всех возможных комбинаций признаков выбирается наилучший в смысле минимаксного критерия ансамбль признаков.

**Описание параметров. Вводимые:**

- $N$  — число признаков;
- $N_1, N_2$  — верхний и нижний пределы интервалов усреднения;
- $N_2$  — число изображаемых 1-го и 2-го классов;
- $F$  — число пропускаемых из ряда в ряд карт разрешаемых спор (свобода выбора);
- $\Theta_1 : N_2$  — вспомогательный массив (используется для заполнения массива  $A_X$ );
- $A_X [1 : N, 1 : N_2]$  — матрица исходных данных.

**Вычисляемые:**

- $\Pi [1 : F, 1 : N, 1 : 2]$  — массив значений порогов  $\Theta_1$  и  $\Theta_2$  для отобранных на первом ряду карт разрешаемых спор;
- $Q [1 : 2, 1 : N]$  — граничные пары эффективных интервалов изменения признаков;
- $X_S [1 : N]$  — средние значения признаков;
- $K [1 : F, 1 : N, 1 : 2]$  — массив характеристики отобранных ансамблей признаков;

**Код [1 : N, 1 : 2]** — массив характеристик наилучшего ансамбля признаков для данного интервала усреднения;  
**ПА [1 : N, 1 : 2]** — лучшее сочетание порогов для всех типервалов усреднения.  
**Последовательность работы программы.** Последовательное заполнение массива  $A_X$ . Вычисление эффективных интервалов изменения признаков и их средних значений. Заполнение массива  $\Pi$ , т. е. выбор  $F$  лучших комбинаций порогов  $\Theta_1$  и  $\Theta_2$ . Составление псевдозадачных карт разрешаемых споров для  $S$  признаков ( $S = 2, 3, \dots, N$ ) и выбор  $NF$  лучших ансамблей.

Выбор на печать полученных результатов: наилучший в смысле минимаксного критерия — ансамбль признаков и соответствующие ему значения порогов, суммарная карта решаемых споров, количество нулей и единиц в суммарной карте разрешаемых споров.

**Пример 7.** Исходные данные представлены в табл. 8. Количество признаков, измеренных в  $N_1 = 40$  точках,  $N = 3$ , свобода выбора  $F = 3$ ; интервалы усреднения варьируются от  $N_21 = 3$  до  $N_22 = 6$ . В результате счета получены следующие данные:

Оптимальный ансамбль признаков: $V_1, V_2, V_3$ .
Сочетание порогов по амплитуде и козябринитеты заполнения пачки: $\Theta_{11}^{11} = 5,7677; \Theta_{12} = 5,7860; \Theta_{13} = 1,9775; \Theta_{21} = 0,3352; \Theta_{22} = 0,3333; \Theta_{23} = 0,6667$ .
Оптимальный интервал усреднения $\Theta_2 = 6$ ; начальная фаза усреднения $F_1 = 3$ ; кратность разрешаемых споров при этом во всех клетках равна 3.

ПРОГРАММА 53

```

1.   'BEGIN' 'INTEGER' FI,F1,C1,U1,P,I,J,T2,02,N,N1,N21,N22,N2,F,
2.   READIN,N21,N22,N2,F),U1-'#2/(2*N21)',,
3.   'BEGIN' 'ARRAY' AK'/(1:N,1:N2)',PA'/(1:N,1:N2)',DA,IK5'/(1:N)/',
4.   W'/(1:N2)',,'INTEGER' 'ARRAY' A'/(1:U1,1:U1)',KA'/(1:N)/',
5.   'FOR P:=1 STEP 1 UNTIL N DO' KA(P):=N,, 'FOR I:=1 STEP 1
6.   'UNTIL' 'N DO' BEGIN' READIN,, 'FOR J:=1 STEP 1 UNTIL N2' DO'
7.   AX(I,J):=1/V(J)',END,, 'FOR Q2:=N21 STEP 1 UNTIL N22' DO'
8.   'FOR PI:=0 STEP 1 UNTIL Q2-1' DO' BEGIN J:=N22-FI,,,
9.   'FOR I:=1 STEP 1 UNTIL N2-2*FI DO' BEGIN J:=J-2*Q2,,,
10.  'IF J>LT 2*Q2 THEN' 'GOTO M1' END,, M1:N1:=I+2*Q2,,,
11.  'BEGIN' 'INTEGER' R,T,G,U,Z,A1,Q1,V,G,U:=R1/Q2,, 'BEGIN'
12.  'ARRAY' X'/(1:N,1:N1)/,PI'/(1:F,1:N,1:2)/,O'/(1:2,1:N/),XS'/(1:N/),
13.  IX'/(1:N1)/,, 'INTEGER' 'ARRAY' M1'/(1:F,1:N,1:U)',Y'/(1:U),
14.  K'/(1:F,1:N,1:N,1:2)',, REALS,S1,S2,S3,, 'FOR' I:=1 STEP 1
15.  'UNTIL' 'N DO' 'BEGIN' 'FOR' J:=1 STEP 1 UNTIL N1/2' DO' X'/(1,J):=
16.  AX(I,FI+J),, 'FOR' J:=N1/2+1 STEP 1 UNTIL N1 DO' X'/(1,J):=
17.  AX(I,FI+2-N1/2+J)'END,, 'FOR' I:=1 STEP 1 UNTIL N'DO'
18.  'BEGIN' O'/(1,I,J):=O'/(2,I,J):=X'/(1,I,J)',XS'/(1,J):=K'/(1,1)/N1,,,
19.  'FOR' J:=2 STEP 1 UNTIL N1 DO' BEGIN' IF' O'/(1,I,J)>T'X'/(1,J) '
20.  'THEN' O'/(1,I,J):=X'/(1,J)' ELSE' IF' O'/(2,I,J)>T'X'/(1,J) '
21.  'THEN' O'/(2,I,J):=X'/(1,J)' ELSE' IF' O'/(2,I,J)>T'X'/(1,J) '
22.  'END,, 'FOR' I:=1 STEP 1 UNTIL N'DO' 'FOR' I:=1 STEP 1 UNTIL N'DO'

```

```

23. 'DO' BEGIN'S:='9*0(/H,I/)+1*XS(/I/),S1:=0(/H,I/)-XS(/I/),,
24. R:=0,J,S1:=1*S1,P:=0,,S2:=S+S1,
25. 'FOR' O1:=P WHILE P<LT' 1*M1 OR 'P GT' 9*M1 DO' BEGIN' P:=0,
26. S2:=S2-S1,, 'FOR' J:=1'STEP'1'UNTIL'N1' DO' IF'X(/1,J/)-S2 GT'0
27. 'THEN' P:=P+1'END'., 'IF'R=0' THEN' BEGIN R:=R+1,,S:=S2+S1.,
28. 'GOTO' J,END'., Q(/H,I/):=S2'END'., 'FOR' H:=1'STEP'1'UNTIL'F' DO'
29. 'FOR' G:=1'STEP'1'UNTIL'N' DO' BEGIN' FOR' P:=1'STEP'1'UNTIL'N
30. 'DO' BEGIN' K(/H,G,P,2/):=U**2.,K(/H,G,P,1/):=0'END'.,
31. K(/H,G,G,1/):=H'END'., 'FOR' I:=1'STEP'1'UNTIL'N' DO' BEGIN'
32. O1:=0,, 'FOR' J:=1'STEP'1'UNTIL'N1' DO' IX(/J/):=X(/1,J/)-
33. 0(/2,I/),,S1:=0(/1,I/)-0(/2,I/),,S2:=1*S1,, 'FOR' S:=S2'STEP'
34. S2'UNTIL' S1'DO' BEGIN' FOR' J:=1'STEP'1'UNTIL'U'DO' BEGIN',
35. Y(/J/):=O., 'FOR' H:=(J-1)*02+1'STEP'1'UNTIL'J*02'DO' BEGIN'
36. IX(/W/):=IX(/W/-S2., 'IF' IX(/W/)>S2., 'IF' IX(/W/),GT,0'THEN'Y(/J/):=Y(/J/)+1
37. 'END'., 'IF' J=1'THEN'R:=T:=Y(/1/)'ELSE' 'IF' R,LT'Y(/J/)'THEN'
38. R:=Y(/J/)'ELSE' IT'T,GT'Y(/J/)'THEN'T:=Y(/J/)'END'., 'FOR' J:=T
39. 'STEP'1'UNTIL'R'DO' BEGIN' IF' J=T' THEN' BEGIN' FOR' H:=1'STEP',
40. -1'UNTIL'U'DO'Y(/H/):=Y(/W/-J'END'., 'ELSE' 'FOR' H:=1'STEP'1
41. 'UNTIL'U'DO'Y(/H/):=Y(/W/-1..,Z1:=0,, 'FOR' H:=1'STEP'1'UNTIL'
42. U/2'DO' 'FOR' G:=1'STEP'1'UNTIL,U/2'DO' Z1:=Z1+(TF,Y(/H/),LE'0
43. ,AND'Y(/G+U/2/)'LE'0'OR'Y(/H/)',GT,0'AND'Y(/G+U/2/),GT,'C
44. 'THEN'1'ELSE'0., 'IF' O1'LT'F'THEN'O1:=O1+1'ELSE'01:=F., 'FOR'
45. H:=1'STEP'1'UNTIL,O1'DO' BEGIN' IF' Z1,LT'R(/H,I,1,2,),THE'
46. 'BEGIN' 'FOR' G:=O1'STEP'-1'UNTIL,H+1'DO' BEGIN' W(/G,I,1,2/):=
47. K(/G-1,I,1,2/),, 'FOR' Z:=1,2'DO' M(/G,I,Z/):=L(/G-1,I,Z/),
48. 'FOR' P:=1'STEP'1'UNTIL,U'DO'M(/G,I,P/):=M1(/G-1,I,P/)'THE'
49. R(/H,I,1,2/):=Z1,,K(/H,I,2/):=(U/2)**2-Z1.,L(/H,I,1/):=S+
50. 0(/2,I/),,L(/H,I,2/):=U/2., 'FOR' P:=1'STEP'1'UNTIL'U'DO'
51. M1,LT,H,I,P/):='IF' Y(/P/)'GT'0'THEN'1'ELSE'0., 'GOTO' J1'END'
52. 'END'., J1:='END' 'END'., P:=U/2., 'BEGIN' INTEGER' ARRAY'
53. K2(/1:F,1:N,1:2/),K1,K3(/1:N,1:2/),M2,M(/1:U,1:U/),
54. 'FOR' P:=1'STEP'1'UNTIL'N'DO' BEGIN' K3(/P,1/):=K(/1,1,P,1/),
55. K3(/P,2/):=K(/1,1,P,2/)'END'., 'FOR' I:=1'STEP'1'UNTIL'N'DO'
56. 'FOR' J:=1'STEP'1'UNTIL'F'DO' IF' K3(/1,2/)',GT'R(/J,I,1,2/
57. 'THEN' 'FOR' P:=1'STEP'1'UNTIL'N'DO' BEGIN' K3(/P,1/):=
58. K(/J,I,P,1/),,K3(/P,2/):=K(/J,I,P,2/)'END'., 'FOR' I:=1'STEP'1
59. 'UNTIL'N-1'DO' BEGIN' V:='IF' I'LT'N'THEN'I+1'ELSE'N',,
60. 'FOR' J:=1'STEP'1'UNTIL'N'DO' BEGIN' O1:=0,, 'FOR' Z:=1'STEP'1
61. 'UNTIL'P'DO' K2(/Z,1,2/):=U**2., 'FOR' G:=1'STEP'1'UNTIL'P'DO'
62. 'BEGIN' 'FOR' P:=1'STEP'1'UNTIL'N'DO' K1(/P,1/):=K(/G,J,P,1/),
63. 'FOR' P:=1'STEP'1'UNTIL'U'DO' 'FOR' R:=1'STEP'1'UNTIL'U'DO'
64. M(/P,R/):=0,, 'FOR' Z1:=1'STEP'1'UNTIL'N'DO' BEGIN'
65. 'IF' K(/G,J,Z1,1/)=0'THEN' GOTO' J4., 'FOR' P:=1'STEP'1'UNTIL'
66. U*2.,DO'Y(/P/):=M1(/K(/G,J,Z1,1/),Z1,P/),, 'FOR' P:=1'STEP'1
67. 'UNTIL'U'DO' 'FOR' R:=1'STEP'1'UNTIL'U'DO' M(/P,R/):=M(/P,R/)+
68. ('IF' Y(/P/)',GT,0'AND'Y(/R+U/)',LE'0'OR'Y(/P/)',LE'0'AND'
69. Y(/R+U/)',GT,0'THEN'0.),J4.'END'., 'FOR' H:=1'STEP'1
70. 'UNTIL'N'DO' BEGIN' IF' K(/G,J,H,1/)',GT,0'THEN' GOTO' J3.,
71. 'FOR' Z:=1'STEP'1'UNTIL'F'DO' BEGIN' IF' J,GT'H'THEN' BEGIN'
72. K1(/H,1/):=Z,, 'FOR' T:=1'STEP'1'UNTIL'J-1'DO' 'FOR' P:=1'STEP'1
73. 'UNTIL'P'DO' BEGIN' Z1:=V., 'FOR' R:=1'STEP'1'UNTIL'N'DO' 'IF'
74. K(/P,T,R,1/)',EQ'R(/R,1/)'THEN' Z1:=Z1-1,, 'IF' Z1'LE'0'THEN'
75. 'GOTO' J6'END' 'END'., 'FOR' Z1:=1'STEP'1'UNTIL'N'DO' K1(/Z1,2/):=
76. 0., 'FOR' P:=1'STEP'1'UNTIL'U'DO' 'FOR' R:=1'STEP'1'UNTIL'U'DO'
77. M2(/P,R/):=M(/P,R/)+( IF'M1(/Z,H,R+U/)',GT,0'AND'M1(/Z,H,R+U/)'
78. ',LE'0'OR'M1(/Z,H,R/)',LE'0'AND'M1(/Z,H,R+U/)',GT,0'THEN'1
79. ',ELSE'0.),T:=-1,, 'FOR' Z1:=1'STEP'1'UNTIL'V'DO' BEGIN' T:=T+1.,
80. 'FOR' P:=1'STEP'1'UNTIL'U'DO' 'FOR' R:=1'STEP'1'UNTIL'U'DO'
81. K1(/Z1,2/):=K1(/Z1,2/)+(IP'M2(/P,R/)=T'THEN'1'ELSE'0)'END'.,
82. 'IF' O1'LT'F'THEN'O1:=O1+1'ELSE'01:=F., 'FOR' P:=1'STEP'1'UNTIL'
83. O1'DO' 'BEGIN' 'FOR' R:=1'STEP'1'UNTIL'V'DO' 'IF' K1(/R,2/)',LT'
84. K2(/P,R,2/)',THEN' GOTO' J5'ELSE' 'IF' K1(/R,2/)',GT' K2(/P,R,2/)'
85. 'THEN' 'GOTO' J9.,J5:'FOR' R:=O1'STEP'1'UNTIL'V'DO' 'IF' K1(/R,2/)',LT'
86. 'STEP'1'UNTIL'N'DO' BEGIN' K2(/R,T,1/):=K2(/R-1,T,1/),
87. K2(/R,T,2/):=K2(/R-1,T,2/)'END'., 'FOR' T:=1'STEP'1'UNTIL'N'DO'
88. 'BEGIN' K2(/P,T,1/):=K(/G,J,T,1/),K2(/P,T,2/):=K1(/P,T,2/),
89. 'END'., K2(/P,R,1/):=Z,, 'IF' V,LT'N'THEN' BEGIN' K2(/P,V+1,2/):=
90. U**2., 'FOR' R:=1'STEP'1'UNTIL'V'DO' K2(/P,V+1,2/):=K2(/P,V+1,2/),
91. -K2(/P,R,2/)',END'., 'GOTO' J6.,J9:='END'., J6:='END'., J3:='END'
92. 'END'., 'FOR' T:=1'STEP'1'UNTIL'F'DO' 'FOR' Z1:=1'STEP'1'UNTIL'N
93. 'DO' BEGIN' K(/T,J,Z1,1/):=K2(/T,Z1,1/),K(/T,J,Z1,2/):=
94. K2(/T,Z1,2/)',END' 'END'., 'FOR' R:=1'STEP'1'UNTIL'N'DO' BEGIN'

```

```

95. K1(R,1):=K(1,1,R,1), K1(R,2):=K(1,1,R,2), END., G:=1.,
96. J:=1., FOR T:=1 STEP 1 UNTIL F DO, FOR P:=1 STEP 1 UNTIL N
98. ,DO, BEGIN, FOR R:=1 STEP 1 UNTIL V DO, IF K1(R,2)/GT,
99. K(V,T,P,R,2):=K(V,T,P,R,2) THEN, BEGIN, GOTO J10, ELSE, IF K1(R,2)/
100. ,LT,K(V,T,P,R,2), THEN, GOTO J17, J10: BEGIN, C:=T, J:=P.,
101. ,FOR H:=1 STEP 1 UNTIL N DO, BEGIN, K(J,H,2):=K(J,T,P,H,2),
102. K1(J,H,1):=K(T,P,H,1), END., , GOTO J17 END., J17: END., T:=0.,
103. ,FOR P:=1 STEP 1 UNTIL V DO, IF K1(P,2)=0, THEN T:=P ELSE,
104. ,GOTO J11, J11: FOR R:=1 STEP 1 UNTIL T DO, IF J13(R,2)=NE0
105. ,THEN, GOTO J12., GOTO J13., J12: FOR P:=1 STEP 1 UNTIL N DO,
106. ,BEGIN, K3(P,1):=K1(P,1), K5(P,2):=K1(P,2), END.,
107. J13: END., , FOR P:=1 STEP 1 UNTIL N DO,
108. T:=P, ELSE, GOTO J10., JAC0:=K1(P,2)=0., , IF K3(P,1)
109. KA(V,R)/NE0 THEN, GOTO J1P., , GOTO J1A, IP:=FOR P:=1 STEP 1
110. ,UNTIL N DO, KA(P,V):=K3(P,V), T2:=G2., CI:=FI, FI:=N1.,
111. ,FOR P:=1 STEP 1 UNTIL N DO, BEIN(K5(P,V):=0., , IF K3(P,1)
112. ,GT,0 THEN, K5(P):=P, END., , FOR R:=1 STEP 1
113. ,FOR P:=1 STEP 1 UNTIL N DO, BEIN(IA(P,R):=0., , IF
114. K5(P,V,1), GT,0 THEN, IA(P,R):=IA(K3(P,1), P, R), END.,
115. ,FOR P:=1 STEP 1 UNTIL U DO, FOR R:=1 STEP 1 UNTIL 2 DO,
116. A(V,P,R):=0., , FOR Z1:=1 STEP 1 UNTIL N DO, BEGIN
117. ,IF K3(Z1,1)=0 THEN, GOTO J14., , K0R,P:=-1 STEP 1 UNTIL U*2
118. ,DO, YU(P):=M1(K3(Z1,1), Z1, P), , FOR P:=1 STEP 1 UNTIL U
119. ,DO, FOR R:=1 STEP 1 UNTIL U DO, A(P,R):=A(P,R)+(1/YU(P))
120. ,GT,0 AND, YU(P+U):=M1(K3(Z1,1), Z1, P+U), GT,0
121. ,THEN, 1 ELSE 0., J14: END., END., END., HA, END.,
122. ,PRINT,, (/ 8ИРИЗНЫМ 2Х,12110), , K5, , FOR I:=1 STEP 1 UNTIL
123. 2*DO, BEGIN, FOR P:=1 STEP 1 UNTIL N DO, K5(P,V):=IA(P,I),
124. ,IF P,EQ,1 THEN, PRINT,, (/,5ИЕТА4, 5X,12(F10.4)), , K5'ELSE
125. ,PRINT,, (/,5ИЕТА2, 5X,12(F10.4)), , K5'END., F:=F1/(2*T2),
126. ,PRINT,, (/ ЭНД2=,15), , F2., , PRINT,, (/ ЭНД1=15), , CI,
127. ,PRINT,, (/,2ЭИМАТИЧА РЕЛАМПХ СПОРОВ), , , FOR I:=1 STEP 1
128. ,UNTIL F,DO, PRINT,, (2016), , FOR J:=1 STEP 1 UNTIL F DO,
129. A(V,I,J), , FOR P:=0 STEP 1 UNTIL N-1 DO, DA(P+1):=P.,
130. ,PRINT,, (/,10ККОМЧСТВО,12X,12110), , DA.,
131. ,PRINT,, (/,12X,12110), , HA, END., END., EOP

```

## ПРИЛОЖЕНИЕ

### ОПИСАНИЕ ПОДПРОГРАММ И ФУНКЦИЙ

#### ПОДПРОГРАММА ОРТИМ1

Решение задач нелинейного (выпуклого) программирования (программа 54).  
*Обращение.* ОРТИМ1 (ХО, М, Н, NO, ПЧ, CQ, G, СШ, ТАУ, Э, EG, Е, F,  
EQUATI, JACOBI).

*Описание параметров:*

ХО — вектор переменных задачи нелинейного программирования;

NO — длина вектора ХО (число переменных);

М — общее число ограничений в виде неравенств и равенств;

Н — сумма числа переменных и числа неравенств;

ПЧ — периодичность вывода на печать промежуточных результатов итерационного процесса поиска экстремума;

CQ — максимальное число итераций;

G — коэффициент, соответствующий при G > 0 поиску максимума, а при

G < 0 — минимума функции;

ШИ — точность выхода шага на каждой итерации;

ТАУ — начальная величина шага;

ЭС — точность соблюдения ограничений;

ЕС — условие останова в случае, когда экстремум находится внутри

области, определяемой неравенствами;

Е — условие останова при экстремуме, находящемся на границе области;

F — значение оптимизируемой функции;

EQUATI — внешняя процедура для вычисления значений функции и вектора

ограничений;

JACOBI — процедура для вычисления значений вектора градиента оптимизи-

руемой функции и матрицы Якоби системы ограничений.

*Требуемые функции и подпрограммы* ПОЛ, ABS, ENTIER.

*Метод.* Решается следующая задача. определить  
 $f^* := \min f(x)$ ,  
где вектор переменных  $x [m_0 \times 1]$  удовлетворяет  $l_0$  ограничениям-равенствам  
 $y_0(x) = 0$  (П.1)  
и  $l_1$  ограничениям-неравенствам  
 $y_{II}(x) \leq 0$ . (П.2)

Задача (П.1) — (П.3) решается с применением дискретного варианта метода дифференциального спуска [3], который является модификацией методов проектирования градиента с восстановлением связей. При этом ограничения (П.3) предварительно стоят к равенствам введением вектора вспомогательных переменных  $\tilde{z}[l_1 \times 1]$ :

$$\tilde{y}_{IIj} = y_{Hj} + \frac{1}{2} (\tilde{z}_j)^2 = 0, \quad j = \overline{1, l_1} \quad (\text{П.4})$$

Определен вектора переменных  $z^T = (x^T, z^T)$  и ограничений-система  
 $y^T(z) = (y_\rho^T(z), y_H^T(z))$  (размерностью  $m = m_0 + l_1$  и  $l = l_0 + l_1$  соответственно),  
 сводим исходную задачу к задаче без первенств:

$$\text{тип } f(z), \quad y(z) = 0. \quad (\Pi.5)$$

Алгоритм решения этой модифицированной задачи является итерационной процедурой:

$$z_{k+1} = z_k + \lambda_k u_k, \quad (\Pi.6)$$

где  $\lambda_k$  — величина шага из точки  $z_k$  в направлении вектора сдвига  $u_k$ . Особенностью применяемого метода является то, что вектор  $u_k$  имеет две ортогональные составляющие: первая  $u^{(1)}$  пропорциональна минимальному расстоянию от  $z_k$  до гиперповерхности  $y(z) = 0$ , а вторая  $u^{(2)}$  — проекции антиградиента функции на эту поверхность. Для вычисления вектора  $u$  составляется неоднородная система линейных алгебраических уравнений:

$$\frac{\partial y_k}{\partial z_k^T} u_k = B_k u_k = -y_k, \quad (\Pi.7)$$

которая решается при помощи процедуры НОА (см. описание) с учетом вектора градиента оптимизируемой функции  $\nabla f$ .

$$u_k = -B_k^+ y_k + \gamma B^0 \nabla f. \quad (\Pi.8)$$

При этом вектор  $u_k$  обеспечивает сдвиг из произвольной точки в направлении допустимой области и условного экстремума функции одновременно ( $\gamma$  соответствует  $G$  программе).

Выбор величины шага осуществляется по алгоритму, реализованному в виде подпрограммы STEP:

$$\lambda_k = \begin{cases} \arg \min_{\lambda > 0} \|y(z_k + \lambda u_k)\|, & \|y_k\| \leq \varepsilon, \\ \arg \min_{\lambda > 0} f(z_k - \lambda u_k), & \|y_k\| > \varepsilon, \end{cases} \quad (\Pi.9)$$

где  $\|y\| = \max_{j=1}^J |y_j|$ . При этом обеспечивается сходимость итерационного процесса (П.6) к условному экстремуму, если соблюдаются каждое из ограничений  $y_i(z) = 0$  с точностью не ниже  $\varepsilon$  (параметр  $\varepsilon$  в тексте программы).

Алгоритм решения задачи типа (П.1) — (П.3), реализованный в виде процедуры ОРГИМ1, осуществляет автоматический переход к модифицированной задаче (П.5) и ее решение по алгоритму (П.6) с регулировкой шага согласно алгоритму (П.9). Данный алгоритм повторяет с небольшими изменениями алгоритм, зафиксированный в Республикалистском фонде алгоритмов и программ (РФАП) [5].

*Примечание.* 1. Управление выводом на печать следующее: при  $\Pi\chi = 0$  вывод на печать следующее: при  $\Pi\chi > 0$  (целом) выводятся результаты на каждой ПЧ-й итерации; при  $\Pi\chi = GQ$  печатается только решение задачи; при  $\Pi\chi < 0$  печатаются также значения вектора ограниченный градиента функции, якобиана ограничений, вектора сдвига и его компонент и др.

2. В процедуре JACOBI должны быть запрограммированы результаты аналитического вычисления частных производных.

#### ПРОГРАММА 54

```

1. 'ALGOT':NOL,
2. 'PROCEDURE'OPTIM(S10,M,N,NO,IN,C0,C1,TAY,J,EG,E,F,ROUTI,
3. JACOB1,'TIEGER1',N,N1,IN,CO.,'REAL6,CM,TAY,3,EG,E,F.,
4. ,ARRAY,X0,,,'PROCEDURE'ROUTI,JACOB1.,
5. ,BEGIN'ARRAY'V/1:M,1:N+M+1/,S/1:M/,A,X,U,U2/1:N+1/,
```

```

6. 'INTEGER'I,J,K,L,O.,'REAL'R,S,C,G1,T,31.,
7. 'PROCEDURE'MAX(X,M,S,I.),'ARRAY'X,,'INTEGER'M,J.,'REAL'S.,
8. 'BEGIN','INTEGER'I.,S:=ABS(X/1/),J:=1.,'FOR'I:=2:STEP1
9. 'UNTIL'M>DO,'IF'ABS(X/I/)>GT'S'THEN'BEGIN'J:=I
10. 'IF'C>T,I'THEN'G2:=R'ELSE'BEGIN'IPR'LE'31'THEN'C2:=-G*F
11. 'PROCEDURE'STEP(X,F,Φ,C),,'ARRAY'X,Φ,,'REAL'F,C.,'BEGIN','REAL'
12. C1,C2,R,T1.,'ARRAY'X1/(1:N+M/),T:=T1:=TAY..C1:=-C2:=-1*F'C*GT*
13. J1'THEN'G'ELSE,-G*F.,M0.'FOR'I:=1'STEP1'UNTIL'N=M1'DO'
14. X1/I/):=X/I/→T*U/I/,,DOUT(X1,F,Φ),,'FOR'I:=1'STEP1
15. 'UNTIL'M1'DO,S/I/):=S/I/→H*0.5*X/I/→H*1*(I+1/)**2,MAX(Φ,M,R,L)
16. 'IF'C>T,I1'THEN'G2:=R'ELSE'BEGIN'IPR'LE'31'THEN'C2:=-G*F
17. 'END',,'IP'T1'LE'C1'AND'T'GT'C1'AND'C2'GT'C1'THEN'BEGIN'T:=T-
    TI.,'GUEOMK'END',,'IP'T1'GT,C1'THEN'BEGIN'IP'G2'LT'C1'THEN'
18. 'BEGIN'IP'T1'IP'TAY,THEN'T1:=T1/2..T:=T+T1..C1:=G2.,'GOTO'N0
19. 'END',ELSE'BEGIN'T1:=T1/2..T:=T-T1,'GOTO'M0'END',,'C:=R.,
20. 'FOR'I:=1'STEP1'UNTIL'N+M+1'DO,X/I/):=X/I/,,EQUAT(X,F,Φ),
21. 'FOR'I:=1'STEP1'UNTIL'N+M+1'DO,X/I/):=X/I/,,EQUAT(X,A,B),,
22. 'FOR'I:=1'STEP1'UNTIL'N1'DO,'BEGIN'X/(N+I/):='TP'
23. 31:=J,,'FOR'I:=1'STEP1'UNTIL'N1'DO,'BEGIN'X/(N+I/):='TP
24. 'Q(I/)'LT'0'THEN'SORT(2*ABS(S/I/)))ELSE'0..1..5/I/):=
25. 'Q(I/)+0.5*X(I+I/)**2'END',,'MAX(Φ,M,C1)..,'FOR'I:=1'STEP1
26. 'UNTIL'N+M+1'DO,R/I/):=0..T:=0..0:=1.,'IP'TH,LTO'THEN'K:=N+
27. M1'ELSE'K=N..M2,'IF'0>CO'THEN'GOTO'KOH,O:=G+1.,'FOR'I:=1
28. 'STEP1'UNTIL'M'DO,'FOR'J:=1'STEP1'UNTIL'N+M+1'DO'B/(I,J/):=
29. 0..R:=-G*F.,'IP'Q'GT'0'THEN'STER(X,F,Φ,C),,'JACOB(X,A,B),,
30. 'FOR'I:=1'STEP1'UNTIL'M1'DO,B/(I,N+I/):=X/(N+I/),,
31. 'FOR'I:=1'STEP1'UNTIL'M'DO'B/(I,N+M+1/):=S/I/)/TAY.,
32. 'IP'P'CH,I'DO,AND,O=0'THEN'PRINT'',/(2X11НКОМАН DY-
33. /(TOM12.4)),,'B,,HO(B,M,N,N+M,Φ,A,G,U,I2),,S:=0.,'IF'C
34. 'LE'31'THEN'BEGIN'IF'-G*F'GT'R'THEN'J1:=31/10..R:=-G1:=0.,
35. 'FOR'I:=1'STEP1'UNTIL'N+M+1'DO,'BEGIN'S:=S+U2/I/)*A/I/),
36. R:=R+U2/I/→U2/I/,,C1:=C1+A/I/→A/I/)'END',,'IF'C1'LE'G
37. 'OR'R=0'THEN'GOTO'KOH'ELSE'S:=S/SQR(R*G1)'END',,'IF'P'CH=0
38. 'THEN'GOTO'MEL,'IP'0/IN=ENTER(0/IN)'THEN'BEGIN'PRINT'',/,
39. 2X,ОШИБКИИ 2X,6ИМЛ T,3Х,6ИМРВЛ,3Х,ННЕВСКА,6Х,
40. 9ICOS(A*U2),6КТФУНИОН),,'IP'K'LE'5'THEN'PRINT'',,(1Н+
41. 75Х,10НЕВЕЧНЫЕ),,'PRINT'',,(2Х,15,4Х,10.3,13,2Х,
```

```

42. 3(4X,E10.3))'',0,T,L,C,S,F,,TR'K,LB'5'THEN'PRINT'',1H,
43. 6BX,5E12.5)),,'FOR I:=1'STEP 1'UNTIL'KDO'X(/I)'ELSE'PRINT'
44. ''(4X,4НЧЕРГАНИЕ X:=(8X10E12.5)),,'FOR I:=1'STEP 1
45. 'UNTIL'KDO'X(/I),,'IF'IN'LT0'THEN'BEGIN'PRINT'',(2X,
46. ЭНDF:=(10E12.4)),,'A.,'PRINT'',(2X2IY/(10E12.4))'',Ф.,
47. 'PRINT'',(2X,8H(U2,U2)=E9.3,5X,8H(DF DF)=E9.3/2X,2HDF=,
48. (10E12.4)),,'R,C1,U,'PRINT'',(2X,3H(U2=/,10E12.4)),,'U2'END',
49. 'END',,ME1:,'IF'C'GT'31'OR'ABS(S),GT'E'THEN'GOTO'ME1..,KOH:
50. 'IF'PC=0'AND'U'LT'CO'THEN'GOTO'ME2..,IF'U=CO'THEN'PRINT',
51. ,,(5X1ИННЕЕ РЕШЕНИЕ),,'ELSE'BEGIN'IF'R=0'THEN'PRINT',
52. ,,(5X1ТЕКСТ. НА ГРАНИЦЕ),,'IF'C1'LE'EG'THEN'PRINT',
53. ,,(5X,16БУКСТВ. В ОБЛАСТИ),,'PRINT',//10X,ЛИЧНО ,
54. 10ИТЕРАЦИИ =,15,3X,7НУРАВН.=,13,ЭХИНЕВЗКА=,E10.3,ЭХ,
55. 10ICUS(DF*U2),E10.3)),,'U1,L,C,S'END',,ME2:,'FOR I:=1'STEP 1
56. 'UNTIL'N'U0'X0(/I):=X(/I)',END',EOP'

```

### ПОДПРОГРАММА НОА

Решение подопределенной системы линейных алгебраических уравнений (с числом уравнений, меньшим числа неизвестных) (прогр. 55).

*Обращение.* НОА (B, M, NO, N, D, A, G, U, U2).

*Описание параметров:*

M — чисто уравнений;

N — чисто неизвестных;

B — матрица решаемой системы, имеющая M строк и N + 1 столбец (и последний столбец записан вектор правых частей с противоположным знаком),

D — вспомогательный вектор, именуемый N + 1 элемент;

A — дополнительный «свободный» вектор размером разностью N;

NO — чисто пневмических элементов вектора A (записанных подряд);

G — коэффициент, учитывающий направление вектора A;

U — вектор общего решения системы;

U2 — вектор общего решения соответствующей однородной системы.

*Требуемые функции и подпрограммы.* Нет.

*Метод.* Общее решение подопределенной системы

$Bu = b$

(П.10)

находится в виде суммы общего решения  $u^{(2)}$  соответствующей однородной системы  $Bu = 0$  и частного решения  $u^{(1)}$  системы (П.10):

$$u = u^{(1)} + u^{(2)}. \quad (\text{П.11})$$

Первая компонента решения определяется с помощью операции псевдообразования

$$u^{(1)} = B+b = B^T(BB^T)^{-1}b, \quad (\text{П.12})$$

а вторая является проекцией произвольного «свободного» вектора  $a$  на гиперплоскость, определяемую пересечением гиперплоскостей (П.10):

$$u^{(2)} = B^Tb = (I - B^T(BB^T)^{-1}B)a. \quad (\text{П.13})$$

При заданном векторе  $a$  решение системы (П.10) единственно. В процедуре НОА формулы (П.12), (П.13) непосредственно не применяются, и матрицы  $B^T$ ,  $B^0$ , а также решение  $u^{(1)}$  в явном виде не вычисляются. Решение  $u^{(2)}$

и и получаются с помощью улучшенной реализации схемы последовательной орто-гонализации, примененной в работе [4].

*Примечания*

1. Вектор  $A$  может быть произвольным, в том числе пустым,
- или равным градиенту заданной функции от переменных.
2. Подпрограмма применима для решения определенных систем, т. е. в общем случае  $M \leq N$ .
3. Матрица  $B$  разрушается

### ПРОГРАММА 55

```

1. 'PROCEDURE'НОА(B,M,NO,N,D,A,G,U,U2)..,'ARRAY'Б,D,A,U,U2..
2. 'INTEGER'Н,NO,N,,REAL'G,,BEGIN'INTEGER'I,J,K,L,,REAL'R,
3. 'ARRAY'C(/1:N+1/),F(/1:N0/),D(/1:/):=0,,FOR'I:=1
4. ,STEP'1'UNTIL'N+1'DO'X(/1/):=M(/1/)+B(/1,I)*B(/1,V),,
5. ,FOR J:=1'STEP 1'UNTIL'M'DO'BEGIN'ARRAY'DL(/1:J-1/),
6. ,FOR'I:=1'STEP 1'UNTIL'J-1'DO'BEGIN'UK(/L/):=0,,FOR'I:=1
7. ,STEP'1'UNTIL'N+1'DO'DL(/L/):=DL(/L/)+B(/J,1)*B(/L,1)
8. 'END',,D(/J/):=1,,FOR'K:=1'STEP 1'UNTIL'N+1'DO'BEGIN'FOR'
9. L:=1'STEP 1'UNTIL'J-1'DO'B(/J,K):=B(/J,K)-DL(/L/)/D(/L/),
10. *B(/L,K),,D(/J/):=D(/J/)+B(/J,K)*B(/J,K)',END',,FOR'
11. J:=1'STEP 1'UNTIL'R+1'DO'BEGIN'C(/J/):=0,,FOR'L:=1'STEP
12. 1'UNTIL'M'DO'C(/J/):=C(/J/)-B(/L,N+1/)*B(/L,J)/D(/L/)',END',
13. ,,C(/N+1/):=1+C(/N+1/),,FOR'I:=1'STEP 1'UNTIL'N'DO'BEGIN'
14. 'FOR'K:=1'STEP 1'UNTIL'NO'DO',BEGIN'P(/K/):=0,,FOR'L:=1'STEP
15. 1'UNTIL'N'DO'P(/K/):=P(/K/)-B(/L,I)*B(/L,K)/D(/L/)',END',
16. ,,IF'I'LE'NO'THEN'P(/1/):=1+P(/1/),,P(/1/):=0,,FOR'K:=1
17. ,STEP'1'UNTIL'NO'DO'U2(/I/):=U2(/I/)+G(P(/L/)-C(/1/)*G(/K/),
18. /C(/N+1/)),,U(/V/):=U2(/V/)+C(/1/)/C(/N+1/)',END',EOP',

```

### ПОДПРОГРАММА MAX

Отыскание максимального элемента заданного вектора (прогр. 56).

*Обращение* MAX (I, F, C, M)

*Описание параметров:*

$\Gamma$  — векторный вектор;

F — разность вектора  $\Gamma$ ;

C — выываемое значение максимального элемента вектора  $\Gamma$ ;

*Требуемые функции и подпрограммы.* Нет.

*Метод.* Последовательное сравнение каждого элемента вектора с наибольшим предыдущим

### ПРОГРАММА 56

```

1. 'PROCEDURE'MAX(X,I,YM,K),,'ARRAY'Х,,REAL'YM,,INTEGER'N,K,,,
2. ,ВСТН',INTEGER'I..YM:=Х(/1/),K:=1,,FOR'I:=2'STEP 1'UNTIL'N
3. ,YM'IF'YM>Х(/I/)'THEN'BEGIN'YM:=Х(/I/),K:=I'END'
4. ,END',EOP',

```

## ПОДПРОГРАММА DES

Перевод десятичного числа в число с другим основанием (прогр. 57).

*Обращение* DES (Q, D, N, S).

*Описание параметров:*

Q — десятичное вводимое число;

D — выводимое соответствующее число (вектор) с другим основанием;

N — размерность вектора D,

S — основание числа D.

*Требуемые функции и подпрограммы:* ENTIER (стандартная).

*Метод:* Элементы вектора D (начиная с N-го) получаются как остаток от последовательного деления нацело числа Q на основание S.

```
1. 'PROCEDURE'DES(Q,D,N,S)..,'INTEGER'Q,N,S..,'INTEGER',ARRAY'D,
2. 'BEGIN' INTEGER'I,C,L,J:=Q.,'FOR'I:=1'STEP,-1'UNTIL'1'DO,
3. 'BEGIN' IF'L'EQ'0'THEN'GOTO'M,C:=ENTIER(L/S),D/C/1/):=L-
4. C S.,L:=C'END',M:'FOR'C:=1'STEP1'UNTIL'1'DO'D/C/1/):=Q'END'
5. 'END',
'EOP,
```

Получение коэффициентов частных описаний (регрессионных моделей) по заданной полной системе нормальных уравнений (прогр. 58).

*Обращение* PAR (D, N, FS, P).

*Описание параметров:*

D — вектор, элементы которого равны 0 или 1;

N — размерность векторов D и P;

FS — вводимая матрица нормальных уравнений с числом строк N и столбцов N + 1, соответствующая полной регрессионной модели;

P — выводимый вектор коэффициентов частной модели GAUSS.

*Требуемые функции и подпрограммы:* GAUSS.  
*Метод:* Формируется умополная (частная) система нормальных уравнений, соответствующая ненулевым (единичным) элементам вектора D, которая решается по процедуре GAUSS.

*Примечание.* Частные модели содержат число членов меньше N, поэтому в векторе P на месте отсутствующих членов записываются пули

## ПОДПРОГРАММА PAR

```
1. 'PROCEDURE'NOS(N,M,ME,MD,Y,FS), 'INFER(N,M,ME,, ARRAY MD,
2. Y,FS.,'BEGIN' INTEGER'I,J,F.,'FOR'I:=1'STEP1'UNTIL'N'DO'
3. 'BEGIN' FS/1,N+1/):=0.,'FOR'F:=M'STEP1'UNTIL'MD'DO'
4. FS/I,N+1/):=FS/I,N+1/)*MD/F,I,1/*Y/F/)'END.,'FOR'I:=1
5. 'STEP1'UNTIL'N'DO',FOR'J:=1'STEP1'UNTIL'N'DO','BEGIN'
6. FS/I,J/):=0.,'FOR'R:=M'STEP1'UNTIL'MD'DO'FS/I,J/)'END
7. FS/I,J/+MD/F,I,J/*MD/F,J/)'END.'END.'EOP'
```

## ПОДПРОГРАММА NOS

Построение матрицы нормальных уравнений по заданной части массивов аргументов и функции (прогр. 59).

*Обращение* NOS (N, M, ML, ID, Y, FS).

*Описание параметров:*

M, ML — число аргументов (членов регрессионной зависимости);

N — начальный и конечный индексы частей массивов аргументов и функции,

ID — вводимый массив (двоичный) значений аргументов;

Y — вводимый массив нормальных уравнений.

*Метод:* Используется метод наименьших квадратов.

*Примечание.* Размерности массивов ID (1 : MM, I : N); Y (1 : MM); FS [1 : N, 1 : N + 1]; MM > ML (MM — число точек наблюдений) задаются в основном программе.

## ПОДПРОГРАММА 59

```
1. 'PROCEDURE'NOS(N,M,ME,MD,Y,FS), 'INFER(N,M,ME,, ARRAY MD,
2. Y,FS.,'BEGIN' INTEGER'I,J,F.,'FOR'I:=1'STEP1'UNTIL'N'DO'
3. 'BEGIN' FS/1,N+1/):=0.,'FOR'F:=M'STEP1'UNTIL'MD'DO'
4. FS/I,N+1/):=FS/I,N+1/)*MD/F,I,1/*Y/F/)'END.,'FOR'I:=1
5. 'STEP1'UNTIL'N'DO',FOR'J:=1'STEP1'UNTIL'N'DO','BEGIN'
6. FS/I,J/):=0.,'FOR'R:=M'STEP1'UNTIL'MD'DO'FS/I,J/)'END
7. FS/I,J/+MD/F,I,J/*MD/F,J/)'END.'END.'EOP'
```

## ПОДПРОГРАММА FUNC

Вычисление значения степенного полинома при заданном значении аргумента (прогр. 60).

*Обращение*. FUNC (A, N1, R, F),

*Описание параметров:*

A — вводимый вектор коэффициентов полинома. Длина вектора равна N1;

R — вводимое значение аргумента;

F — вводимое значение полинома;

N1 — число членов полинома.

*Требуемые функции и подпрограммы.* Нет  
рого расположены по возрастанию степеней переменной, применился схема Горнера.

```
1. SUMOUTINE FUNC(A,N1,R,F)
2. DIMENSION A(N1)
3. R=A(N1)
4. N=N-1
5. DO 1*I=1,M
6. 1 1=N-1
```

## ПОДПРОГРАММА 60

7. 1  $R = R * A(I,I)$

8. RETURN

9. END

**ПОДПРОГРАММА RANG**

Упорядочивает элементы массива **B** в порядке возрастания их величин (прогр. 61).

*Обращение.* RANG (N, B).

*Описание параметров:*

**B** — вводимый и выводимый вектор элементов массива. Длина вектора равна **N**;

**N** — число элементов массива.

*Метод.* Последовательное сравнение величин элементов массива и изменение

порядка их расположения.

**ПРОГРАММА 61**

SUBROUTINE RANG(N,B)

1. DIMENSION B(N)

2. DO 1 I=1,N

3. I=I+1

4. IF (B(I).GT.B(I)) GOTO 3

5. IF (B(I).LT.B(I)) GOTO 3

6. DO 1 J=1,I

7. IF (B(I)-B(J)).LT.0.0001 GOTO 14

8. B(I)=B(J)

9. B(J)=B(I)

10. B(I)=R

11. 1 CONTINUE

12. END

13. END

**ПОДПРОГРАММА ZEROS**

Нахождение действительных корней степенного полинома по методу Мюллера (прогр. 63) [12].

*Обращение.* ZEROS (N1, A, N, C, MAX, EPS1, EPS2, EPS3, ETA).

*Описание параметров:*

**A** — вводимый вектор коэффициентов полинома. Длина вектора равна **N1**;

**C** — выводимый вектор действительных корней. Длина вектора равна **N**;

**MAX** — максимальное число итераций;

**EPS1, EPS2** — абсолютная и относительная точность вычисления корней полинома;

**EPS3** — абсолютная точность распознавания кратных корней;

**ETA** — параметр, определяющий шаг изменения начальной точки для итераций поиска отыскания кратного корня;

**N** — степень полинома плюс единица;

**N1** — степень полинома плюс единица.

*Требуемое подпрограммы и функции.* Нет

*Метод.* Для отыскания корней степенного полинома применяется метод Мюллера [12]. В основу программы положен алгоритм [3], предлашенный для отыскания действительных корней произвольной функции Коэффициенты полинома в массиве располагаются в порядке возрастания степеней переменных.

Рекомендуемые значения некоторых параметров: EPS1 =  $10^{-6}$ , EPS2 =  $= 10^{-18}$ , EPS3 =  $10^{-18}$ , ETA =  $10^{-3}$ .

**ПРОГРАММА 63**

1. SUBROUTINE ZEROS(N1,A,N,C,MAX,EPS1,EPS2,EPS3,ETA)

2. DIMENSION C(N),A(N1)

3. DO 40 K=1,N

4. J=0

5. P=0.9\*C(K)

6. P1=1.1\*K\*(K)

7. P2=C(K)

8. IF (C(K).NE.0) GOTO 20

9. P=-1

10. P1=1

11. P2=C(K)

12. IF (C(K).NE.0) GOTO 20

13. P=-1

14. P1=1

**ПРОГРАММА 62**

1. SUBROUTINE BCOUNT(1S, ID, N)

2. DIMENSION IS(ID)

3. K=1

4. 1 IF(IS(K).EQ.0) GOTO 2

5. IS(K)=0

6. K=K+1

7. IF(K.LE.ID) GOTO 1

8. GOTO 3

9. 2 IS(K)=1

10. 3 K=0

11. DO 4 I=1, ID

12. 4 N=N+IS(I)

13. RETURN

14. END

**ПОДПРОГРАММА BCOUNT**

Увеличение двоичного числа на единицу и определение числа единиц в двоичной записи результата (прогр. 62).

*Обращение.* BCOUNT (1S, ID, N).

*Описание параметров:*

**1S** — вводимый вектор элементов массива, представляющий порядковую запись двоичного числа;

**ID** — разрядность числа.

*Требуемое подпрограммы и функции.* Нет.

*Метод.* Разряды двоичного счетчика в порядке возрастания их старшинства представляются последовательно значениями элементов вектора **1S**. Для увеличения двоичного числа, записанного в **1S**, проверяется состояние первого элемента массива. Если это 0, записывается 1; в противном случае записывается 0, и عملية повторяется для следующего элемента вектора.

```

11.    Y2=0
12.    20 R=P
13.    GOTO 500
14.    1 R=P1
15.    XC=P1
16.    GOTO 500
17.    2 R=P2
18.    X1=F1
19.    GOTO 500
20.    3 XZ=F1
21.    J=-0.5
22.    IF(G(E) 10,50,10
23.    10 I1=-0.1*C(K)
24.    GOTO 200
25.    50 I1=-1
26.    200 E=I+J
27.    T=X0*D**2-X1*E**2+X2*(D+E)
28.    U=T**2-4*X2*D*E*(X0*D-X1*E+X2)
29.    IF (U) 60,60,70
30.    60 U=0
31.    GOTO 71
32.    TO U=SORT(U)
33.    71 V=T+U
34.    V=T-U
35.    1F ((ABS(V)-ABS(W)) 80,80,90
36.    80 U=W
37.    GOTO 94
38.    90 U=V
39.    91 IF (U.EQ.0) U=1
40.    T=-2*X2*E/U
41.    H=T*H
42.    R=R+H
43.    IF ((ABS(H/R)-EPS1) 120,500,500
44.    4 IF (ABS(F1).GE.ABS(X2*10)) GOTO 150
45.    . X0=Y1
46.    X1=X2

```

#### ПОДПРОГРАММА COUNT

Счетчик в системе исчисления с произвольным основанием (прогр. 64).

*Обращение. COUNT(ИТ, К, К1, KON)*

*Описание параметров*

ИТ — неэлиминированный вектор, элементы которого представляют собой

псевдоразрядную запись числа,

К — количество разрядов в числе,

К1 — основание системы исчисления минус единица;

KON — параметр, принимающий значение 1 при переполнении разрядной сетки и 0 — в противном случае.

*Требуемые подпрограммы и функции. Нет.*

*Метод.* Разряды ( $K_1 + 1$ )-ичного числа в порядке возрастания их старшинства представляются последовательностью значащих элементов вектора  $\Pi_T$ . Для увеличения числа на единицу проверяется значение первого элемента массива. Если оно меньше  $K_1$ , то значение прибавляется единица; в противном случае записывается 0, и процедура повторяется для следующего элемента вектора.

```

ПРОГРАММА 64
SUBROUTINE COUNT(IT,K,K1,K01)
DIMENSION IT(K)
K01=0
I=1
IF(K1>IT(I).AND.K1) GOTO 2,
IT(I)=IT(I)+1
GOTO 3
2 IT(I)=1
I=I+1
IF(I.LE.K) GOTO 1
K01=1
3 RETURN
END

```

```

9.      COTO 9
10.     3 IF(K,K01,I) GOTO 5
11.     DO 4 M=I,N1
12.     T=A(L,M)
13.     A(L,M)=A(K,M)
14.     4 A(K,M)=T
15.     5 DO 6 J=L,N1
16.     J=N1+L-J1
17.     6 A(L,J)=A(L,J)/A(L,L)
18.     K1=K+1
19.     DO 7 I=K1,N
20.     T=A(I,L)
21.     L=L+1
22.     DO 7 J=L1,N1
23.     7 A(I,J)=A(I,J)-A(L,J)*T
36.     END

```

### ПОДПРОГРАММА BAL

Вычисление небаланса по заданной системе функций (прогр. 65).

Обращение BAL (B, K, KK, T).

Описание параметров:

B — вводимый вектор значений аргументов и функций баланса. Длина вектора равна K;

KK — число, равное сумме количества аргументов и функций баланса.

K — число используемых функций баланса, по которой вычисляется небаланс.

T — номер функции баланса, по которой вычисляется небаланс.

Требование функции и подпрограммы: FBAL.  
Метод. После обращения к FBAL вычисляется значение функции баланса, получаемое из моделя по K — KK значениям ее аргументов и находится его отклонение от записанного в соответствующей ячейке массива B.

Решение системы линейных алгебраических уравнений (прогр. 65).

Обращение GAUSS(A, N, N1, X, KGA)

Описание параметров:

A — вводимая матрица, содержащая коэффициенты уравнений и элементы вектора правых частей (в N1-м столбце). Размерность матрицы — N \* N1,

N — размерность системы уравнений, N1 = N + 1 (вводимая величина),

X — выводимый вектор решения,

KGA — выводимый параметр решения, принимающий значение 0 при вырожденной матрице и значение 1 — при невырожденной.

Требование функции и подпрограмм: Нет

Метод. Реализован метод исключения Гаусса с обратным ходом

### ПОДПРОГРАММА GAUSS

Решение системы линейных алгебраических уравнений (прогр. 65).

Обращение GAUSS(A, N, N1, X, KGA)

Описание параметров:

A — вводимая матрица, содержащая коэффициенты уравнений и элементы вектора правых частей (в N1-м столбце). Размерность матрицы — N \* N1,

N — размерность системы уравнений, N1 = N + 1 (вводимая величина),

X — выводимый вектор решения, принимающий значение 0 при вырожденной матрице и значение 1 — при невырожденной.

Требование функции и подпрограмм: Нет

Метод. Реализован метод исключения Гаусса с обратным ходом

### ПОДПРОГРАММА 65

```

SUBROUTINE GAUSS(A,N,N1,X,KGA)
DIMENSION A(N,N1),X(N1)
I=0
KGA=1
1 I=I+1
DO 2 K=L,N
    IF(K<L) GOTO 3,2,3,
    2 CONTINUE

```

## ПОДПРОГРАММА RPLACE

Перестановка коэффициентов модели в соответствии с ее структурой (прогр. 67).

*Обращение.* RPLACE (A, N, ID, IS).

*Описание параметров:*

A — вводимый вектор параметров. Длина вектора равна ID;

IS — вводимый вектор, представляющий структуру модели: единицы в IS соответствуют номерам аргументов, вошедшим в модель, а нули — исключенным из модели. Длина вектора — ID;

N — число аргументов, включенных в модель (вводимый параметр);

ID — вводимый параметр, определяющий размерности массивов.

*Метод.* Числа, записанные в N первых ячейках вектора, расставляются на позициях, отмеченных единицами в векторе IS.

### ПРОГРАММА 67

```

1. SUBROUTINE REPLACE(A,N,ID,IS)
2.   DIMENSION IS(ID),A(ID)
3.   K=0
4.   DO 2 I=1, ID
5.   IM=ID-I+1
6.   IF( IS(IM).EQ.1)GOTO 1
7.   A(IM)=0
8.   GOTO 2
9.   1 K=K+1
10.  NI=N-K+1
11.  A(IM)=A(NI)
12.  2 CONTINUE
13.  RETURN
14.  END

```

## ПОДПРОГРАММА COEF

Оценивание параметров линейной модели по исходным данным (прогр. 68).

*Обращение.* COEF (X, Y, NT, NT1, IS, ID, N, A, C, N1, IER, II).

*Описание параметров:*

X — вводимая матрица значений аргументов модели. Размерность матрицы ID \* NT;

Y — вводимый вектор значений выхода модели. Длина вектора равна NT;

NT — количество точек исходных данных (видимая величина);

NT1 — вводимый параметр, определяющий, на какой последовательности нужно применять процедуру метода наименьших квадратов. При NT1 = 0 одинаковые параметры модели находятся по всем NT точкам, при NT1 > 0 —

по первым NT1 + 1 точкам, а при NT1 < 0 — по выборке, включающей ID — число оцениваемых параметров;

IS — вводимый вектор, определяющий структуру модели. Аргументы с номерами, которые соответствуют единицы в IS, включаются в модель, а остальные — исключаются. Длина вектора ID;

N1 — вводимый параметр, N1 = N + 1;

II — вводимый параметр. При II = 0 коэффициенты модели оцениваются по исходным данным, а при II = 1 — по табличным суммам (см. гл. 2);

C — рабочий массив. Размерность массива — N \* N1;

A — выводимый вектор опечаток коэффициентов модели;

IER — выводимый параметр, принимающий значение 1 в том случае, когда модель была успешно построена, и 0 — в противном случае.

*Требование подпрограммы и функции.* GAUSS, RPLACE.

*Метод.* Для оценивания параметров применяется метод наименьших квадратов [30]. Решение систем нормальных уравнений осуществляется по методу Гаусса [12].

### ПРОГРАММА 68

```

1. SUBROUTINE COEF(X,Y,NT,NT1,IS, ID,N,A,C,N1,IER,II)
2.  PARAMETER X(ID,II),Y(II),IS(ID),A(ID),C(N,N1)
3.  B=0
4.  DO 1 I=1,II
5.  T(I)=0          19.  DO 7 J=1, ID
6.  DO 1 J=1,II    20.  IF(T(J).EQ.0) GOTO 7
7.  1 C(I,J)=0      21.  I=I+1
8.  1F(I;T1) 5,2,3  22.  A(I,I)=X(I,1)+A(I,I)*II
9.  2 M=I           23.  C(K,M)=C(K,M)+A(I,K)*B
10. GOTO 4          24.  7 CONTINUE
11. 3 M=I+1         25.  DO 8 J=1,N
12. 4 T2=I T        26.  B(J)=B(J)+A(J,I)*A(I,K)
13. GOTO 6          27.  8 K=1,N
14. 5 M=I           28.  9 CONTINUE
15. 1,3=-M          29.  CALL GAUSS(C,N,N+1,A,IER)
16. 6 DO 9 I=1?M    30.  CALL RPLACE(A,N, ID,IS)
17.  D=Y(I)+B*I    31.  RETURN
18.  I=0             32.  END

```

## ПОДПРОГРАММА INITL

Запись десятичного числа в двоичной форме, каждый разряд которой представляется значением отдельного элемента вектора (прогр. 69).

*Обращение.* INITL (IS, ID, K).

*Описание параметров:*

K — вводимое десятичное число;

IS — выводимый вектор, элементы которого представляют собой поразрядную двоичную запись заданного числа;

ID — число элементов вектора.

*Требование подпрограммы и функции.* BCOUNT.

*Метод.* Обнуление элементов вектора IS и k-кратное прибавление единицы к числу, представленному в IS в двоичной системе исчисления.

ПРОГРАММА 69

```

1. SUBROUTINE INITKIS, ID, K
2. DIMENSION IS(IID)
3. J=1
4. DO 1 I=1, ID
5.   1 IS(I)=0
6. DO 2 I=1, K
7.   CALL BCOUNTIS, ID, J
8.   2 CONTINUE
9.   RETURN
10. END

```

ФУНКЦИЯ SUM

Вычисление суммы элементов некоторого ряда в заданных пределах (прогр. 70).

*Обращение.* A := SUM (I, H, Ш, K, R).

*Описание параметров:*

I — параметр, определяющий элементы ряда;

H — нижний предел суммирования;

Ш — шаг суммирования;

K — верхний предел суммирования;

R — текущее (при данном I) значение элемента суммируемого ряда (любое арифметическое выражение с действительным значением, зависящее от I).

*Требование подпрограмм и функции.* Нет.

ПРОГРАММА 70

```

1. 'REAL' PROCEDURE SUM(I, H, Ш, K, R), 'INTEGER' I, H, Ш, K,
2. 'REAL' R, 'BEGIN' REAL S, I:=H, S:=R,
3. 'FOR' I:=I+1 STEP 1 UNTIL K DO 'IF' S < T 'THEN'
4.   SUM:=S 'END' 'EOP'

```

ФУНКЦИЯ RMA

(*Поиск максимального числа в заданных пределах среди чисел некоторого ряда* (прогр. 71)).

*Обращение.* A := RMA (I, H, K, R).

*Описание параметров:*

I — параметр, определяющий элемент ряда;

H — нижний предел поиска максимального числа;

K — верхний предел;

R — текущее (при данном I) значение элемента анализируемого ряда; любое арифметическое выражение с действительным значением, зависящее от I.

*Требование подпрограммы и функции.* Нет.

ПРОГРАММА 71

```

1. 'REAL' PROCEDURE RMA(I, H, K, R), 'INTEGER' I, H, K,
2. 'REAL' R, 'BEGIN' REAL S, I:=H, S:=R,
3. 'FOR' I:=I+1 STEP 1 UNTIL K DO 'IF' S < T 'THEN'
4.   S:=R, MAX:=S 'END' 'EOP',

```

\* \* \*

В справочнике помещены программы и подпрограммы, разработанные авторами в 1973—1978 гг. К настоящему времени список мог бы быть расширен за счет новых созданных алгоритмов, которые уже не вошли в Справочник. Часть публикуемых программ может быть дополнена или изменена читателем (пользователем) в соответствии с новыми разработками в области самоорганизации математических моделей на ЭВМ (см. журнал «Автоматика», 1979—1980 гг.), в связи с учетом особенностей вычислительных машин. Например, программы, написанные на языке ФОРТРАН-БЭСМ, могут быть использованы на машинах серии ЕС, для чего в них необходимо заменить русские буквы (не совпадающие по написанию с латинскими) на латинские и в головной программе исключить первую карту \*.

Программы данного Справочника предназначены для решения задач идентификации объектов управления, интерполации характеристик и краткосрочного прогноза. Задачи детального (при малом интервале усреднения переменных) и долгосрочного прогноза они не решают.

Для детального и долгосрочного прогноза требуется последовательный отбор (секция) переменных с применением ряда программ (по одной на каждом этапе алгоритма).

На первом этапе обычно рекомендуется программа объектного определения системы управлений, описывающих объект \*\*.

На втором этапе применяется программа гармонического прогноза, описанная в Справочнике (см. гл. 2).

На третьем этапе используется программа перебора долгосрочных прогнозов по критерию баланса переменных (спуск в пространстве структур модельей).

Весь трехэтапный алгоритм реализует новый критерий «одновременного моделирования» на языках, отличающихся по уровню детализации. Установлено, что моделирование на каком-либо одном языке (например, при использовании одного интервала усреднения) принципиально не может обеспечить достаточно детального и долгосрочного прогноза.

Таким образом, программы, помещенные в данном Справочнике, с добавлением двух указанных новых программ могут быть элементами многоэтапной процедуры самоорганизации весьма эффективных (и единственно достойных) долгосрочных сверхдолгосрочных прогнозов.

\* А. А. Пирину. Программирование на алголе и фортране. М., «Наука», 1978.

\*\* Автоматика, 1979, № 1

- СПИСОК ЛИТЕРАТУРЫ**
1. Алгоритм решения задач нелинейного программирования. / Л. М. Бойчук, Н. В. Попков, В. С. Степашко и др. — Киев, РФАП, Ин-т кибернетики АН УССР, 1975, № 214.
  2. Андерсен Г. Статистический анализ временных рядов. М., Мир, 1976.
  3. Библиотека алгоритмов 1б—5б/М. И. Агеев, В. П. Алик, Р. М. Галис и др. М., Сов. радио, 1975.
  4. Бойчук Л. М.— В кн.: Кибернетика и вычислительная техника. Киев, Наук. думка, 1974. Вып. 23.
  5. Бойчук Л. М., Попков Н. В. Ортогональный алгоритм решения пеноопределенной системы линейных алгебраических уравнений. Киев, РФАП, Ин-т кибернетики АН УССР, 1973, № 155.
  6. Высоцкий В. М.— Автоматика, 1976, № 3.
  7. Высоцкий В. Н.— В кн.: Математические модели для прогнозирования и управления качеством водол. Киев, РФАП, Ин-т кибернетики АН УССР, 1976.
  8. Высоцкий В. Н., Ивахненко А. Г., Чеберкус В. И.— Автоматика, 1975, № 1.
  9. Высоцкий В. Н., Тимченко И. К.— В кн.: Программы прямого синтеза моделей (по принципу самоорганизации) Киев, РФАП, Ин-т кибернетики АН УССР, 1975. Вып. 1.
  10. Высоцкий В. Н., Чеберкус В. И.— В кн.: Программы прямого синтеза моделей (по принципу самоорганизации). Киев, РФАП, Ин-т кибернетики АН УССР, 1975. Вып. 1.
  11. Высоцкий В. М., Юнусов Н. И.— Автоматика, 1977, № 5.
  12. Демидович Б. П., Марон И. К. Основы вычислительной математики. М. Физматиз, 1969.
  13. Довгострокове прогнозование видковых процессов за алгоритмами МГВА при критериях неизменности ѹ балансу змінних О. Г. Івахненко, Н. О. Івахненко, В. М. Висоцький та ін.— Автоматика, 1976, № 2, ч. 3; 1976, № 4, ч. 4.
  14. Ивахненко А. Г. Долгосрочное прогнозирование и управление сложными системами. Киев, Техніка, 1971.
  15. Ивахненко А. Г. Системы эвристической самоорганизации в технической кибернетике. Киев, Техніка, 1971.
  16. Ивахненко А. Г., Высоцкий В. Н., Боярко В. П. Алгоритм синтеза моделей сложных объектов для долгосрочного прогнозирования. Киев, РФАП, Ин-т кибернетики АН УССР, 1977, Р. 94.
  17. Ивахненко А. Г., Высоцкий В. Н., Ивахненко Н. А.— Автоматика, 1978, № 1.
  18. Ивахненко А. Г., Зайченко Ю. П., Димитров В. Д. Принятие решений на основе самоорганизации. М., Сов. радио, 1975.
  19. Ивахненко О. Г., Ивахненко Н. О.— Автоматика, 1975, № 4.
  20. Ивахненко Н. О., Кваско М. З.— Автоматика, 1972, № 5.
  21. Ивахненко О. Г., Копна Ю. В.— Автоматика, 1974, № 2.
  22. Ивахненко О. Г., Кротов Г. І.— Автоматика, 1976, № 2.
  23. Ивахненко Н. А., Марчев А. А.— Автоматика, 1978, № 3.
  24. Ивахненко О. Г., Степашко В. С.— Автоматика, 1975, № 5.
  25. Идентификация ѹвязи динамики складного объекта за экспериментальными данными методом самоорганизации моделей / О. Г. Ивахненко, Г. І. Кротов, В. І. Чеберкус та ін.— Автоматика, 1977, № 2, ч. 1.
  26. Казанский О. В., Перчук В. Л., Голиков О. С.— Автоматика, 1975, № 1.
  27. Копна Ю. В.— В кн.: Программы прямого синтеза моделей (по принципу самоорганизации). Киев, РФАП, Ин-т кибернетики АН УССР, 1975. Вып. 1.
  28. Копна Ю. В.— В кн.: Программы прямого синтеза моделей (по принципу самоорганизации). Киев, РФАП, Ин-т кибернетики АН УССР, 1975. Вып. 3.
  29. Копна Ю. В., Ивахненко Н. А., Павлов А. В. Выбор ансамбля информативных признаков на основе принципов самоорганизации. Киев, РФАП, Ин-т кибернетики АН УССР, 1976, Р. 197.
  30. Крамер Г. Математические методы статистики. М., Мир, 1975.
  31. Немчинов В. С. Избранные произведения. В 6-ти т. Т. 1. М., Наука, 1967.
  32. Пека П. Ю., Степашко В. С. Идентификация полей сложных объектов на основе концепции разностной аппроксимации. Киев, РФАП, Ин-т кибернетики АН УССР, 1975.
  33. Про єдиність відновлення кривої регресії за малим числом точок/О. Г. Івахненко, П. І. Ковальчук, М. М. Тодула та ін.— Автоматика, 1973, № 5, ч. 2.
  34. Светильский Б. К., Бойчук Л. М., Мирюков Б. Г.— Автоматика, 1977, № 5.
  35. Степашко В. С. Алгоритм самоорганизации моделей среднемесячного стока рек. Киев, РФАП, Ин-т кибернетики АН УССР, 1977, № 51.
  36. Степашко В. С. Комбинаторный алгоритм МГУА для структурной идентификации объектов управления с применением различных критериев селекции. Киев, РФАП, Ин-т кибернетики АН УССР, 1978, № 4951.
  37. Степашко В. С., Попков Н. В. Алгоритм управления каскадом водохранилищ с оптимизацией по скользящему интервале. Киев, РФАП, Ин-т кибернетики АН УССР, 1976, № 155.
  38. Тоду Н. Н.— В кн.: Программы прямого синтеза моделей (по принципу самоорганизации). Киев, РФАП, Ин-т кибернетики АН УССР, 1975. Вып. 2.
  39. Тоду Н. Н., Гулян Н. В., Светильский Б. К.— В кн.: Программы прямого синтеза моделей (по принципу самоорганизации). Киев, РФАП, Ин-т кибернетики АН УССР, 1975. Вып. 1.
  40. Тришес Ю. П.— Автоматика, 1977, № 4.
  41. Шелудько О. И., Патеруз С. Г. Упрощенный алгоритм идентификации характеристик сложных объектов по МГУА. Киев, РФАП, Ин-т кибернетики АН УССР, 1973, № 117.
  42. Юрачковский Ю. П.— Автоматика, 1977, № 5.

## ОГЛАВЛЕНИЕ

	Стр.
<b>Введение . . . . .</b>	3
<b>Глава 1. Индуктивный метод самоорганизации моделей на ЭВМ . . . . .</b>	5
1. Синтез математических моделей при помощи индуктивного метода самоорганизации на ЭВМ . . . . .	5
2. Три основные структуры алгоритмов самоорганизации моделей на ЭВМ . . . . .	5
3. Необходимость сложности модели . . . . .	7
4. Оптимальное разделение таблицы опытных данных . . . . .	10
<b>Глава 2. Выделение трендов для долгосрочного прогнозирования . . . . .</b>	12
1. Задача долгосрочного прогнозирования по трендам . . . . .	12
2. Программа синтеза моделей для долгосрочного прогнозирования . . . . .	21
Подпрограмма VEER . . . . .	24
Подпрограмма KOMB1 . . . . .	26
Подпрограмма PRI . . . . .	29
Функция FBAL . . . . .	30
Подпрограмма SHIFT . . . . .	30
Функция FINAL . . . . .	31
3. Программа выделения прогнозирующего тренда оптимальной сложности по многоядрному алгоритму МГУА . . . . .	32
4. Программа выделения гармонического тренда с нечротными частотами для прогнозирования колебательных процессов . . . . .	35
Подпрограмма HARMAN . . . . .	38
Подпрограмма WB . . . . .	44
Подпрограмма FRIQ . . . . .	45
Подпрограмма WCOEF . . . . .	47
Подпрограмма WB1 . . . . .	49
5. Программа самоорганизации прогнозирующих моделей циклических процессов . . . . .	50
<b>Глава 3. Структурная идентификация алгебраических и конечно-разностных моделей . . . . .</b>	54
1. Задача структурной идентификации . . . . .	54
2. Программа идентификации характеристик сложных объектов по МГУА . . . . .	56
Подпрограмма FORMXP . . . . .	63
Подпрограмма NEML . . . . .	64
Подпрограмма КОЭФ . . . . .	66
Подпрограмма УУ . . . . .	67
Подпрограмма НУ . . . . .	67
Подпрограмма У . . . . .	68
Подпрограмма GRAF . . . . .	68
3. Программа МГУА с ортогонализацией частных описаний . . . . .	70
<b>Глава 4. Моделирование полей . . . . .</b>	110
1. Задача моделирования физических полей . . . . .	110
2. Программа МГУА с персонализацией переменных для моделирования динамических процессов . . . . .	116
Подпрограмма NM . . . . .	117
Подпрограмма ОП . . . . .	117
Подпрограмма FORM . . . . .	118
3. Программа идентификации динамики полей сложных объектов на основе концепт-разностной аппроксимации . . . . .	124
4. Программа идентификации многомерных полей по перегулярной сетке . . . . .	127
<b>Глава 5. Синтез оптимального управления и исследование устойчивости . . . . .</b>	130
1. Задача оптимального управления и исследования устойчивости . . . . .	130
2. Программа идентификации линеаризованных уравнений динамики спиральных систем и исследование их устойчивости . . . . .	131
3. Программа управления с оптимизацией прогноза линейной дискретной системой с запаздыванием . . . . .	138
4. Программа оптимального управления каскадом волохранилищ с оптимизацией на скользящем интеграле . . . . .	143
<b>Глава 6. Выделение множества признаков и распознавание образов . . . . .</b>	151
1. Применение принципов самоорганизации для оценки минимаксного критерия качества ансамбля признаков . . . . .	151
2. Программа выбора ансамбля информативных признаков на основе принципа самоорганизации . . . . .	157
Приложение. Описание подпрограмм и функций . . . . .	162
Подпрограмма ОРТИ . . . . .	162
Подпрограмма НОА . . . . .	165
Подпрограмма МАХ . . . . .	166
Подпрограмма DES . . . . .	167
Подпрограмма РАР . . . . .	167
Подпрограмма NOS . . . . .	168
Подпрограмма FUNC . . . . .	168

Подпрограмма АГМ . . . . .	73
Подпрограмма ГЕНЕР . . . . .	78
Подпрограмма КОВ . . . . .	78
Подпрограмма ГРАН2 . . . . .	79
4. Комбинаторика для структурной идентификации объектов и процессов управления . . . . .	80
Подпрограмма COMBI . . . . .	82
Подпрограмма REGI . . . . .	83
Подпрограмма ГОД . . . . .	83
5. Программа идентификации сложной системы для оптимального управления (с областями включением заданий аргументов управления)	86
Подпрограмма SEL . . . . .	91
Подпрограмма многорядной селекции с применением комбинаторного перебора частных описаний . . . . .	93
Подпрограмма REGR . . . . .	101
Подпрограмма NORM . . . . .	103
Подпрограмма CORA . . . . .	103
Подпрограмма REGA . . . . .	104
Подпрограмма MISR . . . . .	106
Подпрограмма DICH . . . . .	106
Подпрограмма МЛН1 . . . . .	107
Подпрограмма RAZR . . . . .	107
Подпрограмма ZAR . . . . .	108
Подпрограмма KRIT . . . . .	109
Подпрограмма MISN . . . . .	109
Подпрограмма EKSTR . . . . .	110
Глава 4. Моделирование полей . . . . .	110
1. Задача моделирования физических полей . . . . .	110
2. Программа МГУА с персонализацией переменных для моделирования динамических процессов . . . . .	116
Подпрограмма NM . . . . .	117
Подпрограмма ОП . . . . .	117
Подпрограмма FORM . . . . .	118
3. Программа идентификации динамики полей сложных объектов на основе концепт-разностной аппроксимации . . . . .	124
4. Программа идентификации многомерных полей по перегулярной сетке . . . . .	127
Глава 5. Синтез оптимального управления и исследование устойчивости . . . . .	130
1. Задача оптимального управления и исследования устойчивости . . . . .	130
2. Программа идентификации линеаризованных уравнений динамики спиральных систем и исследование их устойчивости . . . . .	131
3. Программа управления с оптимизацией прогноза линейной дискретной системой с запаздыванием . . . . .	138
4. Программа оптимального управления каскадом волохранилищ с оптимизацией на скользящем интеграле . . . . .	143
Глава 6. Выделение множества признаков и распознавание образов . . . . .	151
1. Применение принципов самоорганизации для оценки минимаксного критерия качества ансамбля признаков . . . . .	151
2. Программа выбора ансамбля информативных признаков на основе принципа самоорганизации . . . . .	157
Приложение. Описание подпрограмм и функций . . . . .	162
Подпрограмма ОРТИ . . . . .	162
Подпрограмма НОА . . . . .	165
Подпрограмма МАХ . . . . .	166
Подпрограмма DES . . . . .	167
Подпрограмма РАР . . . . .	167
Подпрограмма NOS . . . . .	168
Подпрограмма FUNC . . . . .	168

Подпрограмма RANG . . . . .	169
Подпрограмма BCOUNT . . . . .	169
Подпрограмма ZEROS . . . . .	170
Подпрограмма COUNT . . . . .	172
Подпрограмма GAUSS . . . . .	173
Подпрограмма BAL . . . . .	174
Подпрограмма REPLACE . . . . .	175
Подпрограмма COEF . . . . .	175
Подпрограмма INITL . . . . .	176
Функция SUM . . . . .	177
Функция RMA . . . . .	177
<b>Список литературы . . . . .</b>	<b>179</b>