

DOI:
**КОМПЬЮТЕРНАЯ СИСТЕМА ОРГАНИЗАЦИИ ЛИНЕЙНЫХ И ПОЛИЛИНЕЙНЫХ
ОПТИМИЗАЦИОННЫХ РАСЧЕТОВ НА БАЛАНСОВЫХ МОДЕЛЯХ**

Лукацкий А.М., Малахов В.А

*Институт энергетических исследований Российской академии наук (ИНЭИ РАН),
31-2, Нагорная ул., Москва, Россия
macrolab@eriras.ru*

Аннотация. Классифицируются динамические не линейные модели балансового типа. Работа с моделями основана на сочетании методов имитации и оптимизации. Оптимизация допускает как квазидинамический, так и динамический режимы работы. Для динамической оптимизации весьма актуальной является проблема борьбы с накоплением погрешностей в вычислениях.

Ключевые слова: балансовые модели, квазидинамика, полная динамика, динамические связи, нелинейная оптимизация, накопление ошибок.

Введение

Балансовые модели используются в ряде областей, в частности, при изучении сложных экономических систем (экономика страны, промышленность, крупная компания) [1-3], или в многочисленных технических моделях с проблемами совместности условий эксплуатации оборудования, например, в медицинской технике [4-5]. В экономике известны два основных типа балансовых моделей: модель Леонтьева (с квадратной матрицей продукт-продукт [6]), модель Неймана-Гейла (с прямоугольной матрицей отрасль-продукт [7]). Основные типы балансовых моделей приведены на рис. 1.

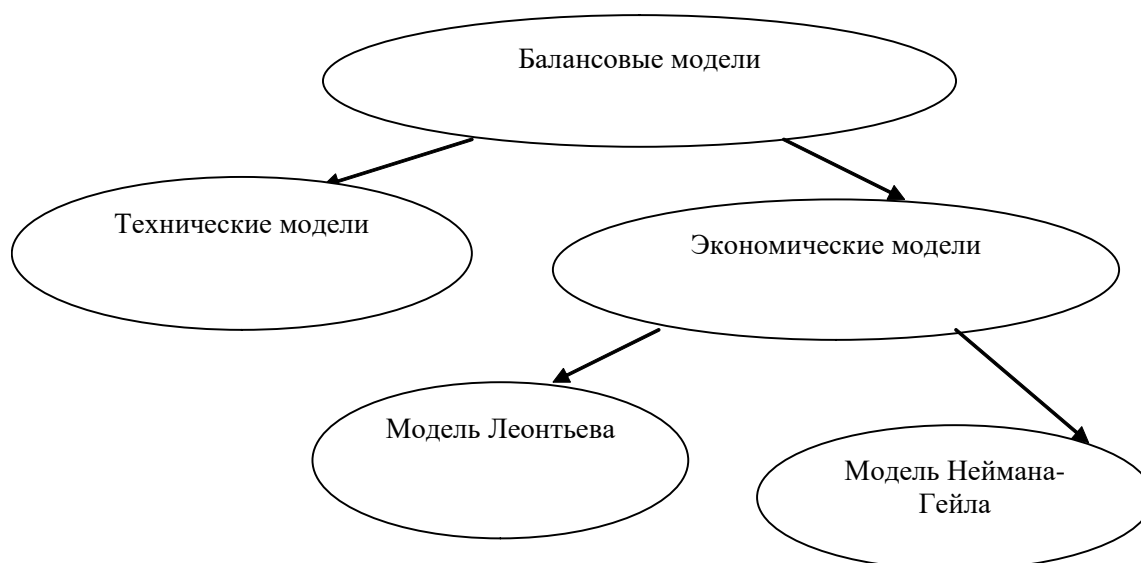


Рис. 1. Типы балансовых моделей.

Процесс разработки балансовых моделей приводит к необходимости моделирования сложных объектов и требует умения работать с различными типами используемых данных. В связи с этим очень полезно иметь гибкую схему для моделирования соответствующих информационных процессов.

В Институте энергетических исследований Российской академии наук (ИНЭИ РАН) были созданы оригинальные инструментальные средства для автоматизированной разработки и эксплуатации оптимизационных и имитационных балансовых моделей типа Неймана-Гейла. Этот подход был реализован в виде программного комплекса (далее именуемого как система).

1 Принципы конструирования моделей балансового типа

Описываемая система включает в себя математические и программные компоненты, которые позволяют специалистам создавать, модифицировать и эксплуатировать широкий спектр моделей балансового типа. Ключевым моментом здесь является то, что это освобождает эксперта от необходимости пользоваться услугами программистов-посредников.

Схема структур данных этой системы показана на рис. 2.

Этот подход предоставляет пользователю следующие возможности:

- простые в эксплуатации средства для редактирования и разработки модели, а также визуального отображения информационных процессов;
- процедуры, включающие активацию алгоритмов прямого подсчета, оптимизации и динамических связей, а также контроль их достоверности;
- шаблоны для ввода данных и отображения результатов расчетов.

За счет возможностей формировать и эксплуатировать модели в он-лайн режиме, когда система дает подсказки пользователю, как действовать в текущей точке процесса, исключаются многочисленные синтаксические ошибки.

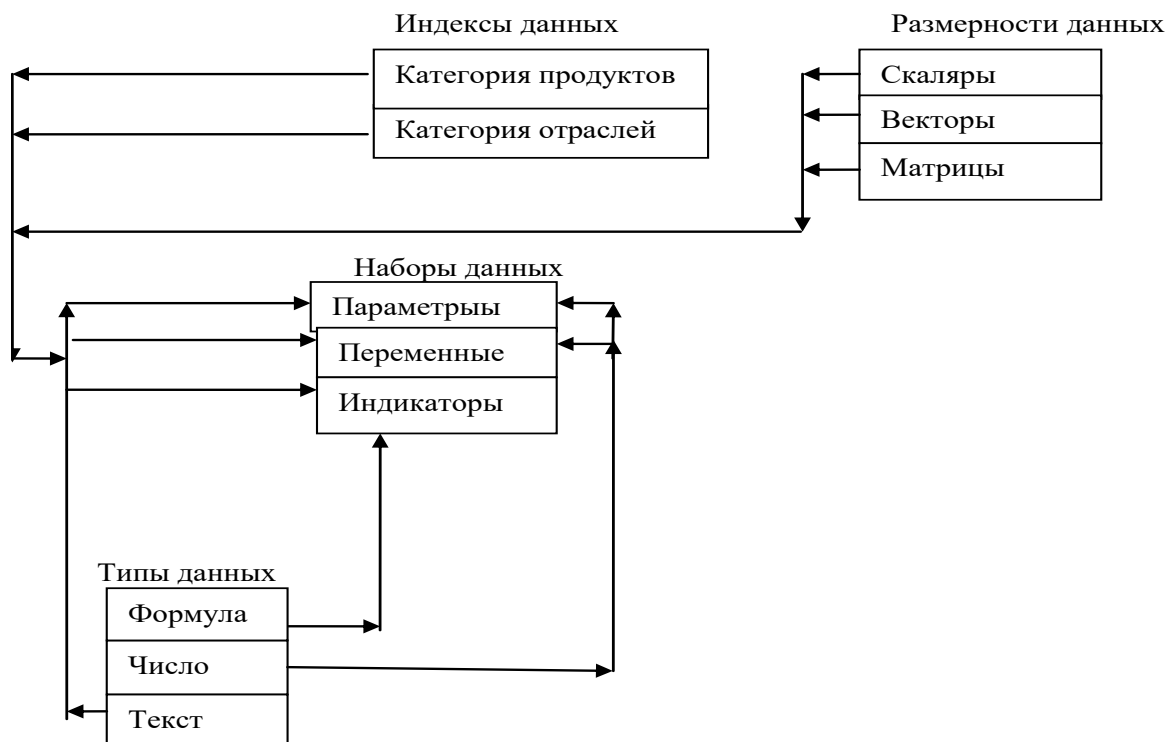


Рис. 2. Диаграмма структур данных системы

2 Средства для ввода и редактирования структуры модели

Для генерации и модификации формальной структуры модели был разработан оригинальный программный комплекс, который обеспечивает интерактивный процесс ввода и редактирования:

- структур данных;
- формул.
- Разрешаются различные типы формул:
- для режима оптимизации – формулы, которые зависят от переменных модели только как полилинейные функции [8];
- для режима имитации – формулы, которые зависят от переменных модели как полиномиальные, так и логические функции.

Система обеспечивает автоматический контроль возвратных ссылок в режиме он-лайн, чтобы избежать заикливания в расчетах.

Пользователю предоставляется интерфейс, который включает в себя пункты меню, подсказки, шаблоны ввода, всплывающие кнопки управления, панели для отображения текущего состояния разрабатываемой модели.

Система работает с моделями балансового типа, описывающих сложные системы, развитие которых представлено в виде последовательности взаимосвязанных временных тактов, как правило, годовых (статические блоки).

Числовыми объектами модели являются:

- экзогенные данные, задаваемые пользователем (параметры);
- данные, получаемые в результате решения задач (переменные);
- показатели, рассчитываемые по значениям параметров и переменных (индикаторы).

Ключевыми атрибутами числовых объектов являются их имена и идентификаторы. Имена используются в заголовках шаблонов и в системных комментариях, а идентификаторы используются в формулах статических блоков и динамических ссылок индексируются членами одной категории, а матричные объекты – двух.

Экзогенные данные устанавливаются в сценарии, и их значения вводятся непосредственно в виде числовых данных (статических параметров) или определяются по формулам динамики, которые определяют их числовые значения для текущего временного цикла в зависимости от значений числовых объектов в прошедших временных циклах.

Переменные – величины, которые доопределяются в задачах математического программирования (оптимизации согласно заданному критерию, поиска допустимых решений).

Индикаторы – величины, значения которых определяются формулами, операндами которых могут быть числа и другие объекты, такие как параметры, переменные и ранее рассчитанные индикаторы. Предлагается удобный интерфейс для ввода формул, которые определяют индикатор как некоторое арифметическое выражение.

Уже введенные объекты допускаются в качестве операндов формулы вводимого индикатора. Формулы состоят из стандартных арифметических выражений и символов суммирования по индексам категорий. Операнды формул индикатора в статическом блоке не имеют временной индексации, то есть относятся только к текущему такту времени. Операнды формул динамической связи имеют такую индексацию, которая относится к прошлым периодам времени. Диалоговое окно для ввода формулы динамической связи показан на рис. 3.

Ввод и редактирование формул в системе выполняется пользователем - непрограммистом. На этапе ввода синтаксис формулы предполагает стандартную структуру арифметических выражений. Процесс ввода ведется в диалоговом редакторе, который предлагает пользователю меню допустимых на текущей точке процедуры ввода операций, причем уведомляет его в случае обнаружения синтаксической ошибки. Такой интерфейс ввода формулы исключает возможность синтаксических ошибок и циклических ссылок. После завершения ввода формула автоматически преобразуется в формат польской (бескобочной) записи. Это ускоряет последующую числовую интерпретацию формулы в расчетном режиме.

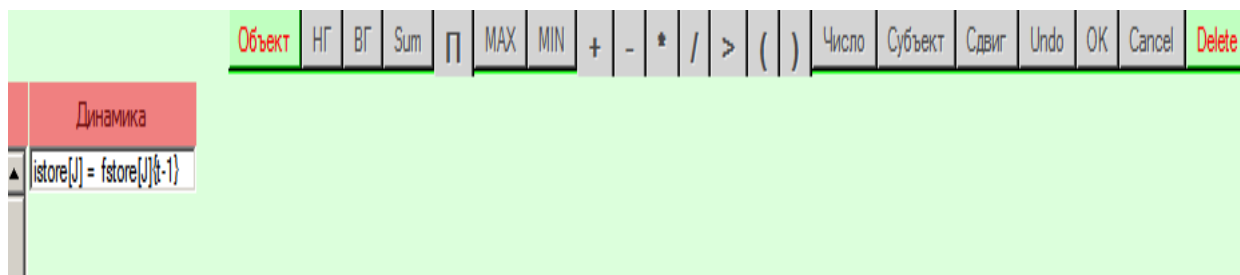


Рис. 3. Диалоговый бокс ввода динамических формул

Арифметические выражения, которые не являются полилинейными, также допускаются в моделях. По запросу пользователя система предоставляет полный список показателей, в формулах которых имеются нарушения полилинейности.

Технология работы с моделью позволяет сохранить ее под другим именем после изменений, внесенных пользователем в состав категорий и объектов. Система предоставляет возможность ведения архива версий модели.

3 Интерактивная процедура ввода данных, запуска режимов имитации и оптимизации

В системе предполагаются следующие режимы:

- ввода динамических ссылок;
- установления требований к расчетам;
- создание шаблонов входных и выходных данных;
- ввода числовых значений входных данных и контроля их полноты;
- проведения прямых расчетов и оптимизации;

- индикации результатов расчетов и их архивирование.

Для каждого нового расчета система автоматически генерирует и предоставляет пользователю набор электронных таблиц на листах Excel. Последовательность листов таблицы соответствует последовательности лет, указанной в спецификации расчета.

Переменные и индикаторы могут иметь опционально верхнюю и нижнюю границы, которые накладывают диапазонные ограничения. Эти границы могут быть статическими или динамическими. По умолчанию для переменных и индикаторов установлены условно бесконечные значения нижней и верхней границ.

Пользователь может вводить числовые значения в специальных шаблонах. Обеспечивается архивирование завершенных расчетов.

В системе предусмотрены следующие режимы расчетов:

- статические вычисления, которые рассчитывают показатели в соответствии с заданными формулами зависимостей, если вы уже ввели значения параметров и базовые значения переменных;
- динамические вычисления (в каждом году, кроме начального), которые вычисляют значения динамических параметров по ранее установленным формулам динамической связи;
- статическая оптимизация;
- динамическая оптимизация.

Оптимизационная процедура предполагает две опции:

- нахождение допустимого решения (удовлетворяющего всем ограничениям модели);
- получение решения, оптимизирующего заданный критерий (показатель, выбранный пользователем из числа скалярных полилинейных показателей модели).

Оптимизация может быть выполнена как для одного временного такта (статическая оптимизация), так и для нескольких последовательных тактов в допустимом диапазоне прогноза (динамическая оптимизация). Для динамической оптимизации критерием является сумма значений выбранного индикатора-критерия на заданной последовательности временных тактов.

Процедуры оптимизации реализуются в рамках задачи полилинейного программирования (ПП) [8]. Для решения задачи ПП был предложен алгоритм, основанный на методе обобщенной релаксации (OMP). Этот алгоритм также использовался в [9] для решения задач, относящихся к сфере CREDAL NETWORKS. Согласно OMP, задача ПП разбивается на серию последовательно выполняемых задач линейного программирования (ЛП). Для реализации OMP были разработаны эффективные программные средства. Они проверяются свойства полилинейности (в частном случае линейности) индикаторов, формируют задачу ПП, а также обеспечивают последовательную генерацию линейных фаз и решение соответствующих задач ЛП.

Состав фаз задачи ПП генерируется автоматически, каждая фаза включает набор переменных, которые образуют линейную подзадачу. Все фазы имеют максимальное наполнение, то есть ни одна фаза не может быть пополнена новыми переменными без нарушения линейности подзадачи, и каждая переменная участвует как минимум в одной фазе. Если, не нарушая полилинейности, можно включить какую-либо переменную в несколько фаз, то эта возможность используется, см. табл. 1.

Прямые расчеты реализуются имитационным режимом работы с моделями. Они могут выполняться независимо от свойств полилинейности индикаторов путем интерпретации формул в статическом и динамическом режимах.

Режим динамической оптимизации подразделяется на квазидинамический и динамический (далее именуемый как полнодинамический).

В квазидинамическом режиме процедура оптимизации запускается последовательно для каждого временного такта t оптимизируемого периода. В этом режиме неполилинейные индикаторы исключаются из задачи статической оптимизации. После завершения оптимизации текущего временного такта t выполняются прямые расчеты. Они показывают, удовлетворяют ли индикаторы указанным для них граничным условиям. Если граничные условия соблюдены, то с помощью динамических связей выполняется переход к следующему временному такту $t+1$ вплоть до исчерпания всего временного периода оптимизации.

В режиме полнодинамической оптимизации статически неполилинейные индикаторы также исключаются, но, кроме того, проверяется, сохраняется ли для статически полилинейных индикаторов свойство полилинейности при переходе к динамике. При обнаружении нарушений полилинейности задача полнодинамической оптимизации объявляется невыполнимой средствами полилинейного программирования. В этом случае пользователю выдается глубина d рекурсии, на которой впервые нарушается свойство полилинейности. После завершения процедуры динамической оптимизации

пользователю выдается сообщение о том, было ли найдено допустимое решение, а также о достигнутом значении критерия, если таковой был специфицирован.

Пример декомпозиции полнодинамической полилинейной задачи на линейные фазы показан в табл. 1. Приведенная задача имеет 3 статических блока, каждый из которых включает 3 линейных фазы. Принадлежность переменной линейной фазе для каждого из статических блоков обозначается символом 1 в номере фазы в блоке. Имеются переменные присутствующие во всех линейных фазах, например, «социальные выплаты (soc)», «государственные субсидии (dot)».

Таблица 1. Декомпозиция полнодинамической полилинейной задачи на линейные фазы

ПЕРЕМЕННЫЕ	статический блок 1			статический блок 2			статический блок 3		
	фаза 1	фаза 2	фаза 3	фаза 1	фаза 2	фаза 3	фаза 1	фаза 2	фаза 3
[1] ИнвестОтр(цвб), Investment[I]	1	0	1	1	0	1	1	0	1
[2] СкрытЗарплОтр(ртг), szo[I]	1	1	1	1	1	1	1	1	1
[3] Дивиденды(ртг), div[I]	1	1	1	1	1	1	1	1	1
[4] Государственные субсидии(ртг), dot[I]	1	1	1	1	1	1	1	1	1
[5] ИмпортПрод(цвб), imp[J]	1	0	1	1	0	1	1	0	1
[6] ЭкспортПрод(цвб), Exp[J]	1	0	1	1	0	1	1	0	1
[7] ИзмЗапПрод(цвб), Chzapp[J]	1	0	1	1	0	1	1	0	1
[8] Соцвыплаты(ртг), soc	1	1	1	1	1	1	1	1	1
[9] СредМесЗарплОтр, srzo[I]	1	1	0	1	1	0	1	1	0
[10] СредМесЗарплГУ, srzg	1	1	1	1	1	1	1	1	1
[11] КоэфПП1, k1[I]	1	0	0	1	0	0	1	0	0
[12] КоэфПП2, k2[I]	1	0	0	1	0	0	1	0	0
[13] ИндексЦены, с[J]	0	1	0	0	1	0	0	1	0
[14] ЗаймыОтр(ртг), Credito[I]	1	1	1	1	1	1	1	1	1
[15] ЗаймыГУ(ртг), Creditg	1	1	1	1	1	1	1	1	1
[16] ИПЦ, CPindex	0	1	0	0	1	0	0	1	0
[17] ИндексРеалПостДоходХ, IncomePersSectorIndex	1	0	1	1	0	1	1	0	1
[18] ВыпускОтр(цвб), vro[I]	0	0	1	0	0	1	0	0	1

Взаимодействие пользователя с системой в случае полнодинамической модели иллюстрируется блок-схемой на рис. 4.

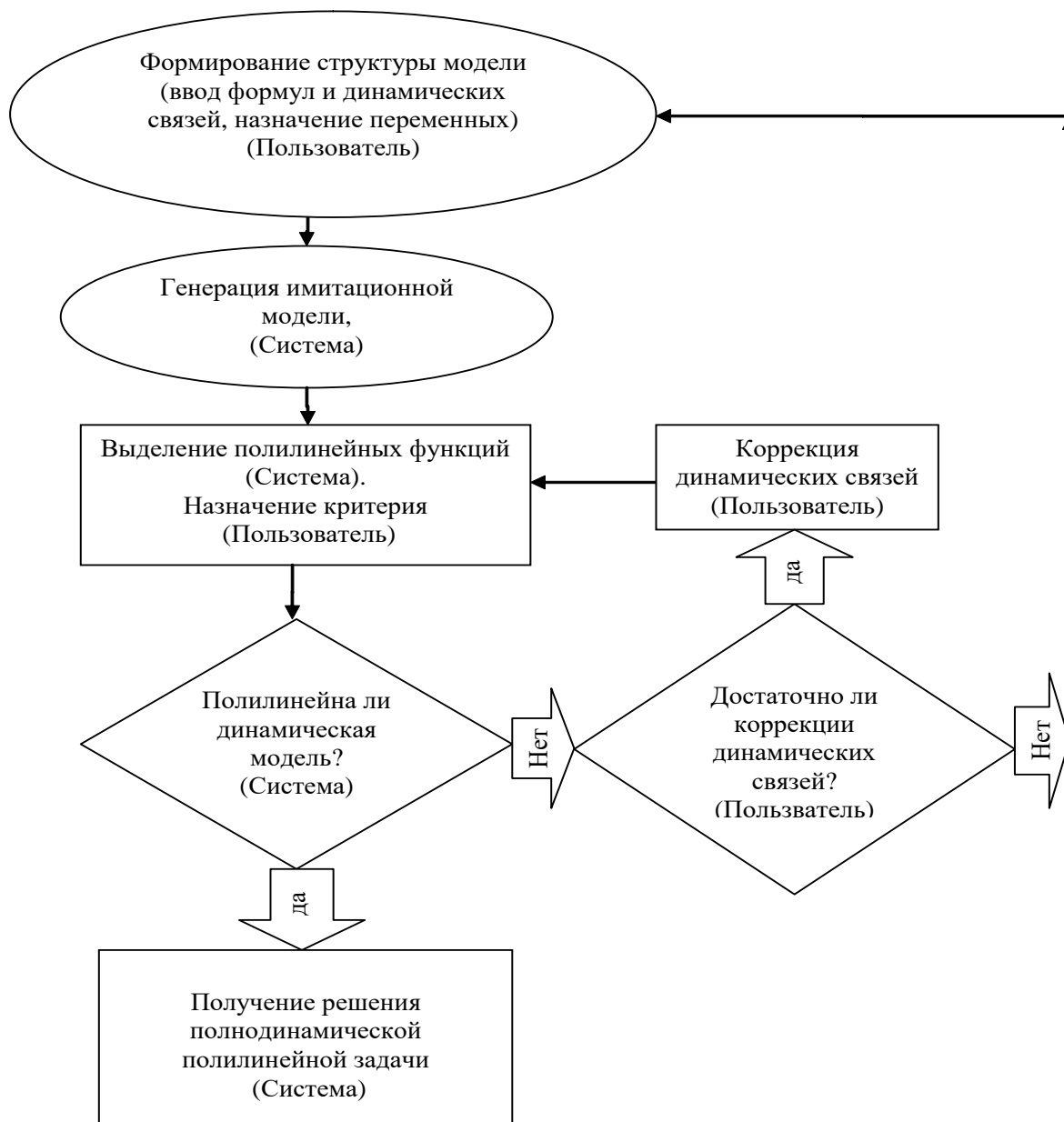


Рис. 4. Взаимодействие пользователя с системой в режиме полнодинамической оптимизации.

В процессе анализа полнодинамической модели на полилинейность проверяются:

- динамические параметры;
- динамические границы переменных и индикаторов.

Проверка ведется в цикле по глубине d ретроспективных динамических связей к текущему такту расчетов t . На глубине d проверяются переменные такта $t-d$. Нарушениями полилинейности являются:

- арифметические неполилинейные выражения, содержащие переменные такта $t-d$;
- логические выражения или специальные неполилинейные функции, включающие переменные такта $t-d$.

Блок-схема процедуры анализа на полилинейность для эксплуатации динамической модели в режиме полнодинамической оптимизации приведена на рис. 5.

Возникновение динамической неполилинейности в статически полилинейной модели иллюстрируется в приложении, пример 1.

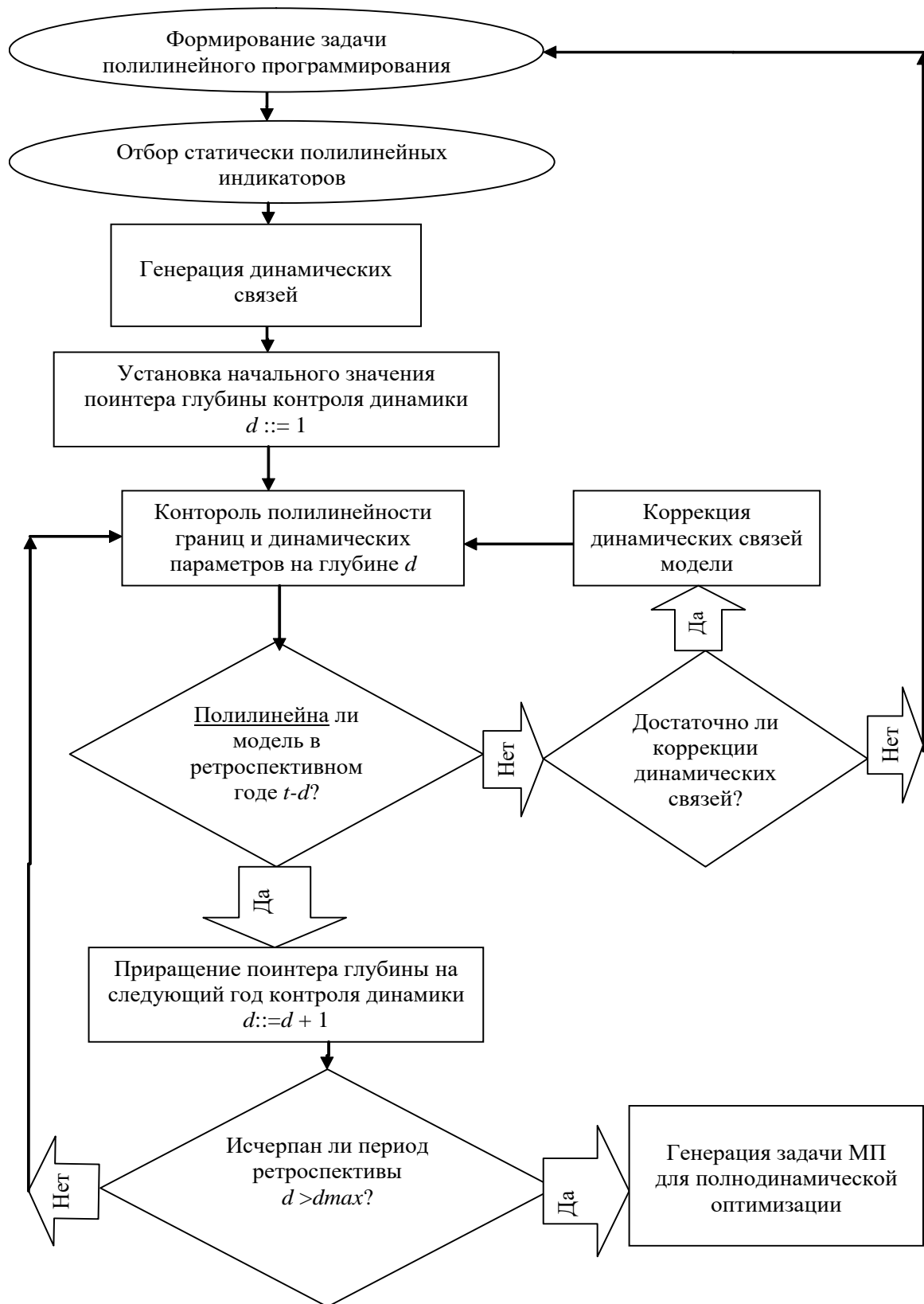


Рис. 5. Блок-схема алгоритма анализа на полилинейность в полнодинамической модели

4 Использование комплекса в процессе эксплуатации макроэкономических моделей

Описанная технология была успешно использована в макроэкономических исследованиях ИНЭИ РАН. Отметим, что, как правило, компьютерные макроэкономические модели используются в качестве «черного ящика» [6], [10]. То есть на входе пользователь указывает определенный набор данных, а на выходе получает строго регламентированный набор макроэкономических показателей.

Напротив, описанный комплекс обеспечил удобную технологию формирования макроэкономических моделей с возможностью их реструктуризации в процессе эксплуатации. Его использование позволило решать такие задачи, как:

- расчет объемов производства и использования продуктов отраслями экономики (межотраслевой баланс);
- оценка энергоемкости валового внутреннего продукта (ВВП);
- определение структуры энергопотребления по отраслям экономики;
- оценка влияния на ВВП факторов внешнего рынка.

5 Проблема накопления ошибок в расчетах

При работе в режиме полной динамической оптимизации особенно острой становится проблема накопления ошибок в расчетах. Это связано с большим количеством исполнений процедуры линейной оптимизации. Основным методом борьбы с накоплением ошибок является повышение точности симплекс-алгоритма [11]. Эффективный метод состоит в том, чтобы осуществить переход от начальной к конечной симплекс-таблице за меньшее количество итераций.

Симплекс-алгоритм, основанный на последовательности шагов жорданового исключения (ЖИ), оказывается здесь наиболее удобным.

Пусть в задаче ЛП общего вида исходная симплекс-таблица имеет n столбцов с переменными, имеющими номера $1, \dots, n$ и m строками, имеющими номера $n+1, \dots, n+m$.

Важно принять во внимание, что если для начальной симплекс-таблицы шаги ЖИ переставляют строчные переменные со столбцовыми, то после нескольких шагов могут переставляться переменные, изначально бывшие только строчными или только столбцовыми. Поэтому введем преобразования $T_{s,r}$.

Если в текущей симплекс-таблице s – строка, r – столбец (или наоборот), то это шаг жорданового исключения;

если в текущей симплекс-таблице s – строка, r – строка (или столбец), то это транспозиция (перестановка) строк (столбцов).

Пусть симплекс-таблица получена из исходной в результате последовательного выполнения преобразований

$$(1) \quad g = T_{s_1, r_1} T_{s_2, r_2} \dots T_{s_k, r_k}$$

Умножение g справа на транспозицию не меняет ЛП-задачу по существу. Поэтому перестановки симплекс-таблицы, приводящие к заданной ЛП-задаче, можно отождествлять с правыми смежными классами

$$S_n \times S_m g \subset S_{n+m}, \text{ где } S_k - \text{симметрическая группа порядка } k, [12].$$

Улучшение точности решения ЛП-задачи можно сформулировать как следующую задачу. Пусть симплекс-таблица первоначального решения получена в результате подстановки g из группы S_{n+m} , которая разложена на шаги ЖИ по формуле (1) и представляет собой слово длины k [12] в смысле системы образующих (2) группы S_{n+m}

$$(2) \quad \{T_{s,r}\}.$$

Требуется найти в правом смежном классе $S_n \times S_m g$ элемента g слово [12]

минимальной длины l

$$\hat{g} = T_{u_1, v_1} T_{u_2, v_2} \dots T_{u_l, v_l}$$

в смысле системы образующих (4).

Алгоритм поиска слова минимальной длины смотрите в приложении, предложение 1.

Машинные эксперименты на крупномасштабной динамической макроэкономической модели показали, что повторный запуск процедуры симплекс-алгоритма с уменьшенным числом шагов ЖИ приводит к повышению точности решения на 3-4 порядка по сравнению с точностью решения, полученного первоначальным запуском симплекс-алгоритма (например, с погрешности $10^{-8} \div 10^{-7}$ до $10^{-12} \div 10^{-11}$).

6 Машинная реализация

Описанный комплекс реализован в Microsoft Visual Studio 2008 под операционной системой WINDOWS-8. Был использован язык программирования Visual BASIC, а для информационной поддержки – MS EXCEL-2010. Все расчеты проводились в среде Visual BASIC. Для увеличения скорости вычислений коллекции Visual BASIC служат временным хранилищем данных.

Программное обеспечение описанной системы оформлено авторскими свидетельствами. В аспекте изучения межотраслевых балансов – [13]. В аспекте ОМР как самостоятельного метода математического программирования – [14].

7 Сравнение с аналогами

Одним из аналогов является MATLAB [15]. Описанное программное обеспечение более удобно и проще в работе специально для класса моделей, описанных в статье, то есть балансового типа, динамических, нелинейных, но полилинейных. MATLAB имеет очевидные преимущества для моделей с более сложной математикой (скажем, описываемых системой дифференциальных уравнений) и необходимостью выдачи результатов в виде пространственной графики, но требует от пользователя специального программирования его задачи средствами процедурного языка системы MATLAB.

Другим аналогом является GAMS [16], который построен по принципу автоматизации программирования, когда под каждую конкретную задачу, описанную на специальном языке, генерируется новая программа, ее реализующая. В отдельных приложениях GAMS объединяют с GEMPACK. В GEMPACK имеется алгебраический язык описания уравнений различных видов (в том числе дифференциальных), но запрещены неравенства. GAMS объединенный с GEMPACK, дает возможность работать с моделями, включающими неравенства [17, с.36], но для этого требуется дополнительное программирование.

Описываемая система поддерживает класс моделей, основанных на системах полилинейных неравенств, которые, как правило, описываются невыпуклыми функциями, для этого класса функций не работают методы поиска равновесных состояний, основанные на выпуклой оптимизации. В описываемой системе применяется принцип прямой интерпретации языка описания моделей. Этот принцип экономит ресурсы компьютера (в крупномасштабных моделях программный код с их описанием может значительно возрасти), позволяет запускать систему в режиме автономной откомпилированной программы без требования наличия каких-либо дополнительных языковых средств на компьютере. Подход, реализованный в описываемой системе, позволяет управлять точностью проводимых расчетов. Последнее становится весьма актуальным в крупномасштабных динамических нелинейных моделях.

Выводы

Построение различных моделей балансового типа средствами описанной выше системы позволяет быстро получать реализации в различных предметных областях. В этой связи, была предложена гибкая схема моделирования. Это позволило внедрить новый тип взаимодействия между разработчиком модели и пользователем.

Более того, пользователи здесь становятся активными участниками процесса реализации модели. Этот подход обеспечивает хорошую основу для исследования конкретной предметной области, в частности, помогает быстро решить проблемы крупномасштабных систем балансового типа. В качестве одного из важных приложений описанная система использовалась для анализа взаимосвязей между топливно-энергетическим комплексом и российской экономикой в целом.

Литература

1. *Malakhov V, Nesytykh K., Dubynina T.*, “A Multi-Agent approach for the intersectoral modeling of the Russian Economy,” 2017 Tenth International Conference Management of Large-Scale System Development (MLSD), IEEE Conference Publications, October 2017, DOI:10.1109/MLSD.2017.8109656.
2. *Lukatskii A.M., Fedorova G.V.*, “Algorithms and software for studying the impact of fuel and energy prices on the economy of the Russian federation,” Management of Large-Scale System Development (MLSD), 2017 Tenth International Conference, 2-4 Oct. 2017, Moscow, Russia, IEEE Xplore, DOI: 10.1109/MLSD.2017.8109653.
3. *Lukatskii A.M., Fedorova G.V.*, “Tools for the Development of Macroeconomic Models of the Fuel and Energy Complex,” 2018 International Multi-Conference on Industrial Engineering and Modern

- Technologies (FarEastCon), 3-4 Oct. 2018, Vladivostok, Russia, IEEE Xplore, DOI: 10.1109/FarEastCon.2018.8602640.
4. *Yadykin I.B., Galyaev I.A.*, "On the Usage of the Energy Functionals for Energy Balance Anomalies Detection of Human Body Organs," // 2018 Eleventh International Conference "Management of large-scale system development" (MLSD), Moscow, 2018, pp. 1-5.
 5. *Ядыкин И.Б., Катаев Д.Е.*, "Оценивание аномалий баланса виртуальной энергии математической модели поджелудочной железы для создания искусственной поджелудочной железы," // Труды 12-й международной конференции "Управление развитием крупномасштабных систем" (MLSD2019), М.: ИПУ РАН, 2019, с.с. 1159-1161.
 6. *Mustafin A., Kantarbayeva A.*, "Opening the Leontief's black box," *Heliyon*. 2018 May; 4(5): e00626, DOI: 10.1016/j.heliyon.2018.e00626++.
 7. *Babaei E., Evstigneev I.V., Schenk-Hoppé K.R., Zhitlukhin M.*, "Von Neumann-Gale Model, Market Frictions, and Capital Growth," (January 13, 2019). Available at SSRN: <https://ssrn.com/abstract=3314852> or <http://dx.doi.org/10.2139/ssrn.3314852>.
 8. *Лукацкий А.М., Шанот Д.В.*, "Методы решения задач полилинейного программирования," Журнал вычислительной математики и математической физики, т. 41, №. 5, 2001, с.с. 680-691.
 9. *Antonucci A., de Campos C.P., Huber D., Zaffalon M.*, "Credal Network Inferences by Linear Programming, in Symbolic and Quantitative Approaches to Reasoning with Uncertainty," Proceedings 12ty Euprean Conference, ESCQARU 2013, Utrecht, The Netherlands, 2013, pp. 13-24.
 10. *Nax H.H., Burton-Chellew M.N., West S.A., Young H. P.*, "Learning in a black-box," *Journal of Economic Behavior & Optimization*, Volume 127, July 2016, p. 1-15.
 11. *Карбовский И.Н.*, "Технология полилинейного программирования в естественно-обусловленных моделях. I," *Автоматика и телемеханика*, 2014, № 9, с.с., 83-96.
 12. *Курош А.Г.*, "Теория групп", 1967, М.: Наука.
 13. *Малахов В.А., Лукацкий А.М., Шанот Д.В.*, "Модель взаимосвязей энергетики с экономикой (МЭНЭК)," Сертификат государственной регистрации компьютерной программы, No. 2018611677 5 февраля, 2018.
 14. *Лукацкий А.М., Шанот Д.В.*, "MULTILINPROGRAM, " Сертификат государственной регистрации компьютерной программы, No. 2018619802, 13 августа, 2018.
 15. *Moler Cleve*, "The Growth of MATLAB and The MathWorks over Two Decades". News & Notes News letter. MathWorks.(January 2006). Retrieved August 14, 2013.
 16. *Rutherford T. F.*, "Applied General Equilibrium Modeling with MPSGE as a GAMS Subsystem: An Overview of the Modeling Framework and Syntax", *Computational Economics*, 14, 1999, 1-4. doi:10.1023/A:1008655831209.
 17. *Kohlhaas M., Pearson K.R.*, Introduction to GEMPACK for GAMS Users, April 2002. [//www.researchgate.net/publication/5007906_Introduction_to_GEMPACK_for_GAMS_Users](http://www.researchgate.net/publication/5007906_Introduction_to_GEMPACK_for_GAMS_Users).

Приложение

Пример 1. Пусть в статически полилинейной модели имеются динамические скалярные параметры p , q , векторный параметр $s[i]$; статические векторные переменные $x[i]$, $y[i]$ и скалярная переменная z .

Для текущего временного такта t заданы динамические связи:

$$p[t] = \max_i (x[i]\{t-2\});$$

$$q[t] = \frac{z\{t-2\}}{z\{t-3\}};$$

$$s[i][t] = \text{abs}(y[i]\{t-2\}).$$

Тогда статически полилинейная модель будет динамически полилинейной на глубине 1, но у нее обнаружится динамическая неполилинейность на глубине 2. В этом случае полнодинамическая оптимизация возможна на периодах $\{t \div t + 1\}$ и невозможна на $\{t \div t + 2\}$.

Пример 2. Пусть симплекс-таблица имеет 10 столбцов с переменными с номерами 1, 2, ..., 10; 10 строк с номерами 11,12, ..., 20.

а) после последовательного выполнения шагов $T_{1,12}, T_{1,11}$ (подстановки $g = T_{1,12} T_{1,11}$) получается симплекс-таблица с номерами столбцовых переменных 12,2,3,...,10 и строчных 1,11,13,...,20;

б) после выполнения $T_{1,11}, T_{1,12}$ (подстановки $h = T_{1,11} T_{1,12}$) – симплекс-таблица с номерами столбцовых переменных $11, 2, 3, \dots, 10$ и строчных $12, 1, 13, \dots, 20$, откуда $g \neq h$

Для случаев а) и б) наборы столбцовых и строчных переменных итоговой симплекс-таблицы отличаются и, следовательно, задают неэквивалентные ЛП-задачи

Предложение 1. Алгоритм улучшения точности решения ЛП-задачи следующий.

Шаг 1. Выделим в симплекс-таблице решения ЛП-задачи, полученного в результате подстановки g вида (2), следующие подмножества:

- столбцов, полученных перенесением строчных переменных исходной симплекс-таблицы;
- строк, полученных перенесением столбцовых переменных исходной симплекс-таблицы.

Числа этих строк и столбцов должны быть равны, обозначим их через w .

Шаг 2. Вычислим списки кодов столбцов и строк подмножеств из шага 1 и упорядочим их по возрастанию. Пусть они имеют вид $\{p_1, \dots, p_w\}$ для столбцов, $\{q_1, \dots, q_w\}$ для строк.

Построим подстановку по элементу $g \in G$ следующим образом

$$(5) \quad \hat{g} = T_{q_1, p_1} T_{q_2, p_2} \dots T_{q_w, p_w}.$$

Заметим, что в разложении (5) используется обратный порядок следования кодов в парах строк и столбцов. Симплекс-таблица, полученная после подстановки \hat{g} подстановки может отличаться порядком следования строк и столбцов по отдельности, поэтому \hat{g} не совпадает с g , принадлежит правому смежному классу $\hat{g} \in S_n \times S_m g$.

Шаг 3. Реализуем перестановку строк и столбцов в исходной симплекс-таблице, задаваемую разложением (4).

Шаг 4. Итоговая симплекс-таблица по составу кодов строк и столбцов совпадает с симплекс-таблицей первоначально полученного решения. Получаем из нее новое оптимальное решение ЛП-задачи. Длина w слова (4), реализующего полученное улучшенное решение ЛП-задачи, минимально возможная, т.к. для перехода к симплекс-таблице первоначально полученного решения ЛП-задачи надо переставить, по крайней мере, w столбцов со строками.

В примере 2а первоначальной подстановкой будет слово длины 2: $g = T_{1,12} T_{1,11}$,

словом минимальной длины в смежном классе $S_{10} \times S_{10} g$ будет слово длины 1: $\hat{g} = T_{1,12}$;

в 2б первоначальная подстановка – слово длины 2: $h = T_{1,11} T_{1,12}$,

словом минимальной длины в смежном классе $S_{10} \times S_{10} h$ – слово длины 1: $\hat{h} = T_{1,11}$.