

О ДВУХ НОВЫХ ПРОГРАММНЫХ СИСТЕМАХ ДЛЯ ФОРМИРОВАНИЯ И АНАЛИЗА ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

© 1994 г. Медницкий В.Г., Медницкий Ю.В.

(Москва)

Описываются две программные системы, позволяющие использовать на РС (АТ-286, 386) суперкальк (SC-4) для подготовки данных и формирования больших задач линейного и частично целочисленного программирования (с последующим анализом их оптимальных планов), когда для решения задачи применяется один из стандартных пакетов математического программирования с исходным файлом в MPS-формате

Впервые модель линейного программирования сформулирована и изучена Л.В. Канторовичем [1, 2] в связи с определенной задачей управления производством. К концу 1940-х – началу 1950-х годов было обнаружено много аналогичных проблем в других сферах деятельности и осознано, что все разнообразие конкретных задач укладывается в каноническую форму [3–5]

$$\min\{cx \mid Ax + Ey = b, \quad x, y \geq 0\},$$

где A, c, b – матрица структурных переменных и векторы коэффициентов целевой функции и правых частей; E – псевдоединичная матрица, позволяющая сформировать произвольные типы ограничений; x, y – векторы структурных и дополнительных переменных. Примерно в это же время были разработаны основные алгоритмы [5] решения этой задачи, но широкое использование нашли лишь варианты модифицированного симплекс-метода с мультипликативным представлением обратной матрицы [6, 7].

При проведении расчетов довольно быстро выяснилось, что в большинстве практических задач доля ненулевых элементов незначительна (в крупной задаче $1E3 \times 1E4$ – меньше 1%). Отсюда как будто бы следует и решение держать в памяти ЭВМ лишь ненулевые элементы с их координатами. Однако простые комбинации типа j, i, f (номера столбца и строки, значение коэффициента), очевидно, трудно интерпретируемы. Полные же имена сущностей, стоящих за безликими номерами i, j с сохранением всех необходимых атрибутов, могут оказаться весьма длинными. Поэтому фирма IBM пошла на компромисс, разрешив использовать 8-символьные имена с теми ограничениями, которые существуют при образовании имен файлов. Описываются лишь не равные нулю элементы A, b, c , но можно одновременно указать несколько функционалов и правых частей со своими именами, т.е. матрицы $B: C$, вместо соответствующих векторов, а вместо E – типы ограничений. В результате возникла структура MPS-файла [8], в основе которой лежит строка перфокарты с описанием элементов данных по определенным форматам. Обязательные секции файла:

NAME – содержит имя задачи,

ROWS – перечисляются имена всех функционалов и ограничений с указанием типов последних,

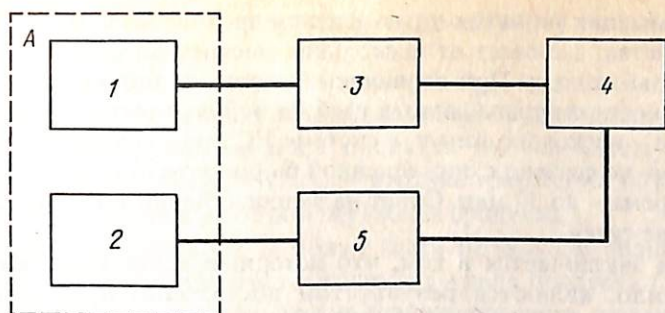


Рис. 1. А – Интегрированная система; 1 – система подготовки данных; 2 – система анализа и представления результатов; 3 – генератор входного файла; 4 – фирменный пакет оптимизации; 5 – система обработки выходного файла

COLUMNS – включает ненулевые элементы матрицы $[A^*, C^*]^*$ с предшествующими именами столбца и строки,

RHS – состоит из ненулевых элементов B с указанием имен правой части и ограничения.

В качестве необязательной файл может содержать секцию верхних (нижних) границ на переменные, начинающуюся с имени BOUNDS и описывающую тип границы, ее имя, имя переменной и значение.

Хотя структура MPS-файла первоначально создавалась для ввода с перфокарт, оказалось, что непосредственное общение с ЭВМ на этом уровне невозможно. Даже небольшую задачу нужно сначала представить на бумажном носителе, а затем очень тщательно перенести ненулевые элементы с указанием всех необходимых имен на бланки перфокарт по точному формату полей. Эта работа выполнима лишь для матриц небольшого размера.

Таким образом, чтобы воспользоваться методом линейного программирования, надо иметь систему поддержки в конфигурации рис. 1. (Пакет математического программирования, конечно, фирмой закрыт и взаимодействие с ним возможно лишь на входе и выходе.)

Временные характеристики системы, как показывает опыт решения многих задач, лимитируются работой оптимизационного пакета. На персональном компьютере, например АТ-286, можно решать сравнительно небольшие задачи, содержащие до 800 ограничений и 1200 переменных. Однако в зависимости от структуры матрицы A уже при 300–400 уравнениях и 500 переменных время решения колеблется от нескольких минут до 2–3 ч непрерывной работы. (При наличии 386-го процессора появляется возможность решения задач большой размерности.) Количество итераций довольно близко соответствует известной из практических расчетов оценке $2m-3n$, но время одной итерации существенно зависит от структуры задачи.

С началом эксплуатации все практические задачи, как правило, довольно быстро усложняются и увеличиваются в размерах. Типичная задача включает 1–1,5 тыс. ограничений и 10–40 тыс. переменных. В системе типа ЕС-1045 решение такой задачи в среднем занимает от 2 до 4 ч непрерывной работы ЭВМ, хотя бывают "рекорд-смены", требующие до 20 и даже 40 ч. На практике это означает, что возможно решение задачи в оперативном режиме с временем реакции 1–2 сут. Время работы прочих элементов системы обычно существенно меньше.

Хорошая система подготовки данных имеет принципиальное значение. Неупорядоченный и "ручной" сбор информации по наблюдениям за работой аспирантов, а также ряда организаций, проводивших в ГВЦ Госплана СССР в 1960–1980-е годы реше-

ние задач оптимизации развития и размещения производства в различных отраслях народного хозяйства, занимает от нескольких месяцев до года. Этот этап аналогичен загрузке базы данных. При вариантных расчетах, подобных работе в режиме "запрос-ответ", корректировка данных требует небольшого времени. На персональном компьютере – несколько минут, в системе ЕС при решении больших задач – до нескольких часов, но связано с определенной бюрократией в прохождении информации; реальное время – до 30 мин. Ответ на запрос обычно получается в течение дня или на следующие сутки.

Проблема же заключается в том, что исходные данные оптимизационной задачи, как правило, являются результатом достаточно простой, рутинной, но трудоемкой обработки некоторого набора первичных данных. Таким образом, система подготовки данных (СПД) оптимизационной задачи в идеале должна быть элементом, причем очень небольшим, некоторой интегрированной системы более общего назначения, подготавливающей входные файлы генератора по запросу в заданной структуре и форматах. Одновременно для не очень больших задач должна существовать удобная форма непосредственного ввода данных через какое-нибудь достаточно распространенное средство. Мы выбрали суперкальку в версии SC4, поскольку именно в этом варианте достигается оптимальное сочетание между возможностями системного средства и ресурсами памяти, которые остаются у пользователя для размещения данных. Таким образом, на входе генератора оказываются файлы суперкальку, которые могут создаваться либо непосредственно, либо в СПД. В файлы того же типа система размещает и результаты решения задачи. Пользователь может обработать его непосредственно средствами SC4 или передать для последующей обработки в систему анализа и представления результатов САПР (рис. 1). Следовательно, можно решать задачи во взаимодействии с интегрированной средой, если затраты на ее разработку оправданы, или только с использованием возможностей SC4.

1. КОНЦЕПЦИЯ СИСТЕМЫ TRAG

Система TRAG (Traffic of Agreement Goods) позволяет решать весьма разнообразные задачи, связанные одной общей идеей получения экономической оценки совокупности предложений по заключению контрактов на приобретение партий товаров для последующей их реализации с учетом условий приобретения, транспортировки, перепродаж при наличии собственных средств или необходимости обращения к заемному капиталу.

Перед Марко Поло стояли относительно простые задачи: продавая свои товары, он мог не спеша собрать сведения, где, что и почем, определить следующий пункт маршрута, приобрести наиболее выгодный набор товаров с учетом всех обстоятельств, таких, как безопасность пути, поборы местных властей, стоимость перевозки и т.д. Перед современным Марко Поло по существу стоят те же задачи, но условия их решения стали более сложными. Главное из них – фактор времени. Не умея быстро оценить степень выгоды того или иного контракта, современный Марко Поло рискует потерять его. Однако и неправильные решения недопустимы, так как могут приводить к большим убыткам. Система TRAG стремится уберечь его от ошибок, причем позволяет принять оптимальное решение не по одному, а сразу по большой совокупности контрактов, как уже заключенных, так и находящихся в стадии обсуждения.

Решение принимается с помощью математической модели, оперирующей с наборами продуктов $i \in I$ и пунктов $l \in L$, связанных сетями различных средств перевозки $n \in N$. Каждая сеть – ориентированный граф, дуги которого описываются множеством упорядоченных пар (l, s) , $s \in S_n$, $l \in L_n \subset L$, где L_n – вся совокупность пунктов (вершин) сети, через которые может проходить средство перевозки n ; S_n – под-

множество пунктов этой сети, в которые можно перейти по некоторой дуге, выходящей из l .

С сетью связываются следующие параметры: d_{ilsn} – дальность при перевозке продукта i из пункта l в пункт s средством перевозки n по кратчайшему маршруту; c_{ilsn} – затраты при перемещении единицы продукта по указанному маршруту; e_{ilsn} – коэффициент потерь, $0 \leq e_{ilsn} < 1$; a_{ilsn} – верхняя граница на объем перевозки продукта; b_{ilsn} – нижняя граница на объем перевозки продукта.

Подмножество вершин $L_1 \subset L$ может быть объявлено возможными пунктами приобретения продуктов. Соответственно: p_{il} – цена продукта i , установившаяся в пункте l ; c_{il} – дополнительные затраты на единицу продукта, связанные с его приобретением; a_{il} – верхняя граница возможного объема закупок; b_{il} – нижняя граница возможного объема закупок; e_{il} – коэффициент потерь.

Подмножество вершин $L_2 \subset L$ может быть объявлено возможными пунктами реализации продуктов. С этим видом деятельности связывается еще одно понятие способа реализации $m \in M$. Поэтому аналогичные величины приобретают дополнительный индекс: p_{ilm} – продажная цена продукта i в пункте l при реализации по способу m ; c_{ilm} – дополнительные затраты на единицу продукта, которые могут возникнуть при его реализации; a_{ilm} – верхняя граница на объем реализации; b_{ilm} – нижняя граница на объем реализации; e_{ilm} – коэффициент потерь.

Ряд пунктов может оказаться вершинами нескольких сетей; произвольное их подмножество L_0 объявлено пунктами перевалки, через которые продукты получают возможность переходить с одной сети на другую. Параметры, описывающие этот процесс, аналогичны предыдущей группе. За ними закреплено значение индекса $m = 0$.

Допустимый объем перевозок с помощью средств определенного типа также ограничен, причем здесь необходимо учесть зависимость величины транспортных услуг как от объемов перевозимых грузов, так и от оборачиваемости подвижного состава, зависящей от дальности перевозок. В соответствии с этим a_n – максимально возможный объем грузооборота с учетом всего доступного парка средств перевозок типа n . (Если объем грузов измеряется в тоннах, а дальность в километрах, то a_n должно определяться в $t \cdot \text{км}$.)

Далее будем предполагать (это выглядит достаточно реалистичным), что все операции по приобретению продуктов и доставке их к пунктам реализации оплачиваются в моменты совершения соответствующих сделок, а экономический эффект возникает после реализации. Таким образом, принимая решение, наш Марко Поло должен принять во внимание величину начального капитала для покрытия расходов из собственных или заемных средств, так как в последнем случае в расходы войдут проценты за кредит. В модели этот механизм описывается двумя параметрами: a – собственный капитал, который может быть выделен для оплаты всех сделок, и E – ожидаемое отношение всей суммы процентов к привлекаемому заемному капиталу.

Переменные модели – это оптимальные объемы соответствующих операций. Они включают: x_{ilsn} – объем перевозок продукта i из пункта l в пункт s средством перевозки n ; x_{il} – объем приобретения продукта i в пункте l ; x_{ilm} – объем реализации продукта i в пункте l по способу m (перевалки, если $m = 0$); u – свободный остаток собственного капитала; v – объем привлекаемых заемных средств.

Математическая модель описывается условиями

$$a_{il} \geq x_{il} \geq b_{il}, \quad (1)$$

$$(1 - e_{il})x_{il} + (1 - e_{il0})x_{il0} - \sum_{s,n} x_{ilsn} = 0, \quad (2)$$

$$\sum_{s,n} (1 - e_{isln}) x_{isln} - \sum_m x_{ilm} - x_{il0} = 0, \quad (3)$$

$$\sum_{i,l,s} d_{ilsn} x_{ilsn} \leq a_n, \quad (4)$$

$$a_{il0} \geq x_{il0} \geq b_{il0}, \quad (5)$$

$$a_{ilsn} \geq x_{ilsn} \geq b_{ilsn}, \quad (6)$$

$$a_{ilm} \geq (1 - e_{ilm}) x_{ilm} \geq b_{ilm}, \quad (7)$$

$$v - \sum_{i,l} \{ (p_{il} + c_{il}) x_{il} + (p_{il0} + c_{il0}) x_{il0} + \sum_m c_{ilm} x_{ilm} + \sum_{s,n} c_{ilsn} x_{ilsn} \} - u = -a, \quad (8)$$

$$\min \left\{ (1 + E)v - u - \sum_{i,l,m} p_{ilm} x_{ilm} \right\}. \quad (9)$$

В (1) устанавливаются ограничения на объемы приобретения продуктов; (2) означает, что вывоз из любого пункта складывается с учетом потерь из объемов закупок и дополнительного количества продуктов, появившегося после перевалки. В соответствии с (3) весь объем ввезенного продукта за вычетом потерь после перевозок распределяется между способами его реализации и перевалкой. В (4)–(6) ограничиваются объемы транспортных услуг по видам средств перевозок, перевалок и перевозок на коммуникациях; в (7) с учетом дополнительных возможных потерь введены ограничения на фактические объемы реализации, а в (8) и (9) заложен несложный экономический механизм: поскольку при $E > 0$ в оптимальном решении выполняется условие $uv = 0$, то при $v > 0$ к части затрат, превышающей собственный капитал, добавляется выплата процентов.

В массивах данных допускается неполное описание. В связи с этим приняты следующие соглашения по умолчанию:

не указана верхняя граница – соответствующее ограничение в наборах (1), (5)–(7) не формируется;

не указана нижняя граница – соответствующая переменная не принимает отрицательных значений;

значения обеих границ указаны и совпадают – величина переменной фиксируется на их общем значении;

значение коэффициентов $d, \dots, e, \dots, c, \dots$ принимается равным нулю;

пользователь не определяет a_n ; вместо соответствующего ограничения в (4) формируется неравенство

$$\sum_{i,l,s} d_{ilsn} x_{ilsn} \geq 0, \quad (10)$$

значение его левой части в оптимальном решении выводится в файле $\langle \text{Result} \rangle$; пользователь не указывает значения a и условия (8), (9) принимают вид

$$v - \sum_{i,l} \left\{ (p_{il} + c_{il}) x_{il} + (p_{il0} + c_{il0}) x_{il0} + \sum_m c_{ilm} x_{ilm} + \sum_{s,n} c_{ilsn} x_{ilsn} \right\} = 0, \quad (11)$$

$$\min \{ v - \sum p_{ilm} x_{ilm} \}; \quad (12)$$

значение для E не запрашивается, оптимальные значения v и функционала (12) выводятся в файле $\langle \text{Result} \rangle$;

если a определено, то необходимо указать положительное E ; оптимальное значение (9) увеличивается на a и выводится в файле $\langle \text{Result} \rangle$; пользователь в обоих случаях видит оптимальное значение прибыли (с переменной знака).

2. ПРЕДСТАВЛЕНИЕ ДАННЫХ В TRAG

Система может использоваться в двух режимах: автономном и Import. В первом база данных организуется в трех линейных файлах SC4: ЗАКУПКИ – английский эквивалент: <Purchase>; ПРОДАЖИ – <Sale>; ТРАНСПОРТ – <Transport>.

Содержание файлов понятно из их заголовков. Заголовки фиксированы (защищены от записи) и размещаются в блоках A1 : H5, A1 : I5, A1 : I5. Блоки A6 : H9999, A6 : I9999, A6 : I9999 по крайней мере теоретически могут содержать записи соответствующих таблиц. В наших обозначениях структура записи такова:

<Purchase>: *l i b a p c e x*

количество символов: 1, 3,

<Sale>: *l i m b a p c e x*

количество символов: 2, 3, 1,

<Transport>: *l n s i b a d c e x*

количество символов: 2, 1, 2, 3,

причем в шифрах в качестве символа разрешено брать цифры или буквы.

В автономном режиме количественные показатели могут представляться значениями или формулами их вычисления, расположенными в клетках таблиц, соответствующих месту показателя. При этом разрешается любое (санкционированное) использование пространства полотна суперкалька справа от последнего столбца соответствующей таблицы. Кроме того, при ее заполнении можно без каких-либо ограничений применять все операции SC4 [9]. Последняя позиция записи, отмеченная символом *x*, при подготовке данных получает любое значение или остается пустой.

Относительно режима Import предполагается, что тело таблицы создается в какой-то другой системе и по команде <Import> передается в файл SC4.

3. УПРАВЛЕНИЕ ПРОЦЕССАМИ ОБРАБОТКИ ДАННЫХ В TRAG

Управление TRAG осуществляется через систему меню с обычными правилами установки курсора в нужный пункт, <Enter> подтверждает выбор, а выход – по <Esc> или <F10>. Появление на экране главного меню означает, что TRAG готова воспринимать указания. При первом входе в TRAG необходимо выполнить <Setup>. Эта команда позволяет установить текущую конфигурацию системы с помощью появляющегося на экране меню (см. рис. 2). Необходимо указать пути к:

<Work directory> – рабочей директории, в которой будут размещаться файлы CAL и MPS, и системные файлы TRAG. PRB, TRAG. CFG;

<TRAG directory>, в которой находится система TRAG;

<LP83 directory>, где находится файл LP83. EXE, и <Supercalc directory>, в которой находится файл SC4. COM.

Выполняя <Setup> повторно, можно внести изменения при получении окна с текущей директорией. Команда <Help lang> позволяет изменить язык подсказки, а <Sound> выводит на экран последнее меню этого пункта, как показано на рис. 2. Оно дает возможность устанавливать (отменять) музыкальное сопровождение соответствующих этапов работы системы с выбором одной из двух мелодий.

Остальные пункты главного меню имеют следующий смысл: <New model> вводит имя новой задачи: не более 7 символов, допускаются знаки: A–Z, 0–9, –, .;

<Delet model> удаляет задачу из системы, сохраняя ее информационные файлы с расширением CAL;

<Quit> – выход в DOS;

<Old model> – основная команда, позволяющая создавать и редактировать файлы исходных данных средствами SC4.

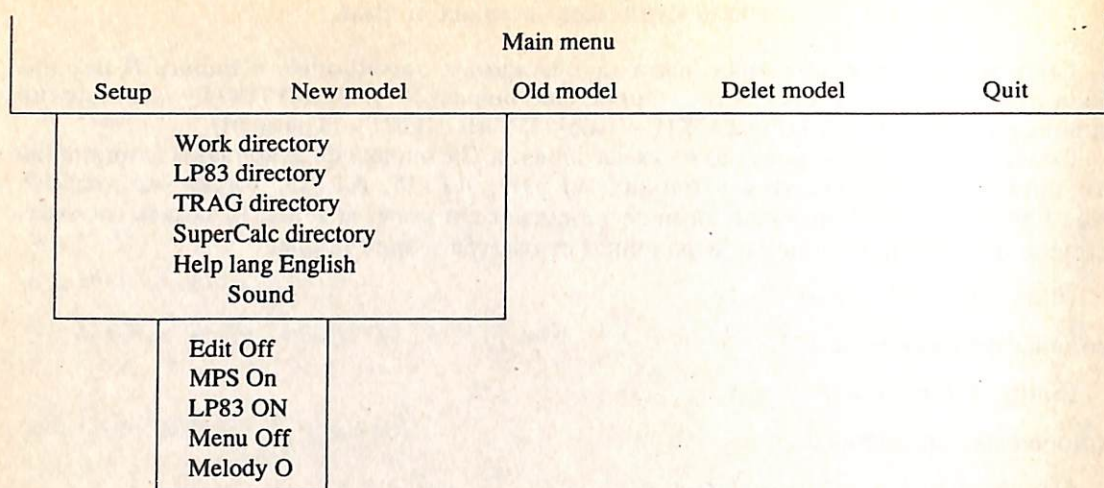


Рис. 2

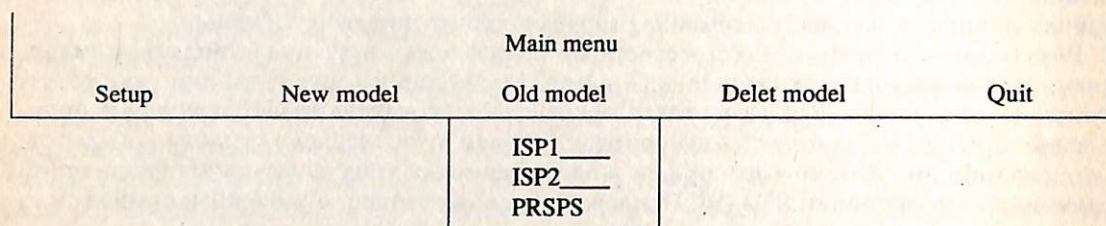


Рис. 3

После выполнения этой команды на экране, как показано на рис. 3, возникает список задач, которые в данный момент поддерживаются системой. Выбрав одну из них, можно попасть в следующее меню (рис. 4). Выбор одного из пунктов:

⟨Purchase⟩, ⟨Sale⟩, ⟨Transport⟩ приводит к вызову следующего меню (рис. 5);

⟨Creat⟩ позволяет начать формировать таблицу, выбранную на предыдущем шаге. Если файл уже существует, то вся находившаяся в нем информация будет потеряна. Таким образом, при повторном обращении к файлу данных выполняется ⟨Edit⟩;

⟨Import⟩ вводит в таблицу информацию из файлов с именем ⟨имя | задачи⟩ ⟨P | S | T⟩ CSV и возвращает в предыдущее меню.

Файл CSV является точным отображением соответствующего файла CAL, но не должен включать его заголовка. Таким образом, доступ к файлу открывается ⟨Edit⟩.

Выполнение этой команды перемещает пользователя в систему SC4 при одновременной загрузке выбранной таблицы. При ее редактировании можно пользоваться всеми средствами SC4, в частности, по окончании сеанса необходимо сохранить таблицу. Однако после / Q, Y, TRAG возвращает в предыдущее меню (см. рис. 4), и теперь можно либо перейти к редактированию любой из двух других таблиц, либо по ⟨Esc⟩ уйти в главное меню и выйти из системы, т.е. информационные файлы создаются в нескольких сеансах работы ЭВМ. Закончив разработку файлов данных, можно выполнить команду ⟨End Edit⟩, выводящую в меню, показанное на рис. 6.

⟨MPS⟩ – команда на формирование MPS-файла задачи.

По окончании этой работы TRAG возвращается в меню (рис. 6) и команда ⟨LP83⟩ выводит на экран меню, приведенное на рис. 7:

⟨Fieldsize⟩ – устанавливает размер числового поля;

⟨Decimals⟩ – количество знаков после десятичной точки;

⟨Alternate⟩ – номер альтернативного решения, результаты которого будут выводиться в отчет;

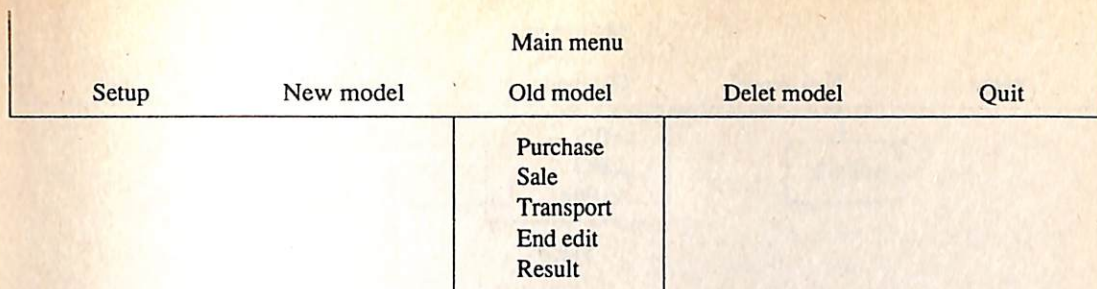


Рис. 4

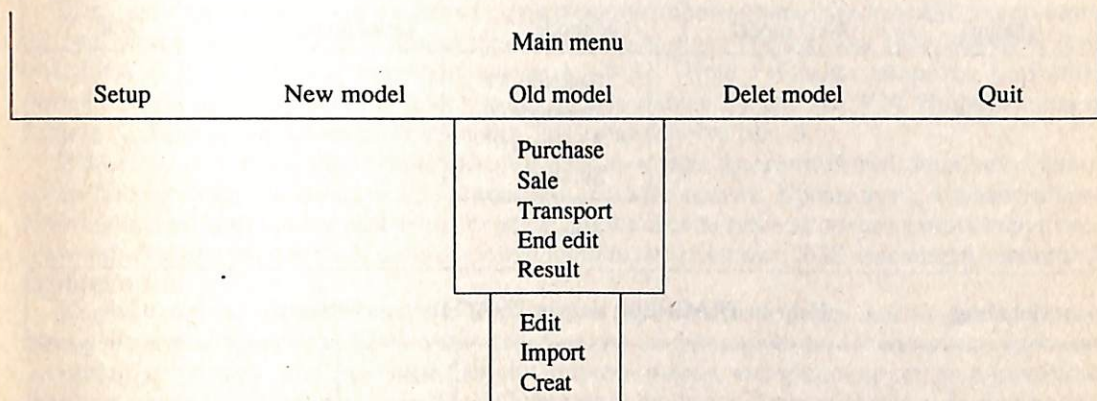


Рис. 5

⟨Solution⟩ – команда на запуск пакета.

Когда решение задачи окончено, TRAG помещает полученные результаты в позицию x файлов входных данных, формирует файл общих результатов ⟨Result⟩ и возвращается в предыдущее меню (рис. 6). Теперь выполнение команды ⟨to Edit⟩ приведет к возврату в меню (рис. 4). Команда ⟨Resalt⟩ вызывает файл с общими характеристиками оптимального решения (с переходом к меню на рис. 5 и ⟨Edit⟩).

В процессе работы в TRAG к любому пункту каждого меню по ⟨F1⟩ можно вызвать подсказку, интерпретирующую пункт и содержащую небольшую инструкцию. С текстами их, а также с общим описанием TRAG можно ознакомиться дополнительно, запустив файл ⟨Traghelp⟩.exe. В результате появления меню, приведенного на рис. 8, с помощью ⟨Help lang⟩ – можно изменить язык подсказки; ⟨Help on TRAG⟩ – вызвать общее описание системы TRAG; ⟨Help on Menu⟩ – вызвать меню, в котором можно вызвать любую подсказку по меню системы TRAG; ⟨Quit⟩ – выход в DOS.

Таким образом, инструкция пользователю становится очень простой. Закончив подготовку файлов данных, выполняются команды ⟨End Edit⟩ и ⟨MPS⟩, даются ответы на вопросы машины и после возвращения в меню – ⟨LP83⟩. В появившемся меню достаточно выполнить команду ⟨Solution⟩, хотя можно предварительно перестроить командную строку LP-83, но это требует некоторого знакомства с пакетом и через несколько минут в позиции x файлов данных, к которым можно обратиться по команде ⟨to Edit⟩, будут указаны рекомендуемые оптимальные размеры товарных партий.

Команда ⟨Result⟩ позволяет получить общие характеристики решения: прибыль со знаком минус, остаток собственного капитала с признаком U , если он образовался, или потребность в дополнительном капитале с признаком V (весь объем необходимых расходов, если в диалоге с машиной нет надобности использовать заемный капитал). С признаком GZ выводится размер собственного капитала (со знаком минус). В следующих строках файла находятся указанные в диалоге с машиной ограничения на средства транспорта и рекомендуемые объемы их использования (если эти ограничения не

Main menu				
Setup	New model	Old model	Delet model	Quit
		MPS LP83 to Edit		

Рис. 6

Main menu				
Setup	New model	Old model	Delet model	Quit
		Fieldsize 11 Decimals 06 Alternate 01 Solution		

Рис. 7

Main menu				
Help lang	Help on TRAGHELP	Help on TRAG	Help on Menu	Quit
		Main Setup Sound Problem Edit Edit/2 MPS LP83		

Рис. 8

формировались, то потребности по видам средств перевозок). Дальнейшая обработка результатов решения не предусмотрена. В автономном режиме ее можно выполнить средствами SC4, в <Import> экспортировать файлы данных или любую их часть в интегрированную среду.

4. УНИВЕРСАЛЬНЫЙ ГЕНЕРАТОР GLPP2

Модель (1)–(10) относится к типу унифицированных, описывающих некоторый класс задач в признаке к данным, организованным в определенных структурах. Поэтому как только изменение в содержательной постановке задачи приводит к данным иной структуры, система сразу же становится недееспособной. Однако, используя наиболее общую форму представления данных с помощью прямоугольной таблицы, мы сталкиваемся с еще более трудными проблемами, поскольку их структуризация перекладывается на пользователя.

Генератор задач линейного и частично целочисленного программирования GLPP2 – это компромисс между стремлением обеспечить условия для изменения типа модели непосредственно в ходе расчетов и необходимостью в максимальной степени упростить процесс подготовки данных при использовании современных пакетов математического программирования. Генератор создает полную версию MPS-файла, включая цело-

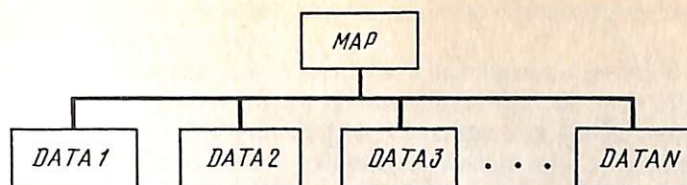


Рис. 9

численные переменные и SOS-множества, а также переменные с фиксированными и свободными значениями. Секция RANGES не поддерживается.

Для данного генератора в отличие от известных промышленных изделий этого типа не требуется специального языка для описания данных (таких, как PDS/MAGEN или GAMMA [8]). В GLPP2 задача создается в SC4. Поле таблицы является рабочим пространством пользователя и могут использоваться все средства SC4. Информация о задаче представляется на двух уровнях, как показано на рис. 9.

В MAP – карте задачи – описывается модель в виде клеточной таблицы текстовых имен блоков модели с именами их координат в заголовках. Кроме того, в поле таблицы глобально определяются типы ограничений и имена блоков специальных структур, в которых содержатся описания целочисленных переменных, SOS-множеств, значений границ и т.д.

Карта модели может многократно редактироваться и лишь после выполнения пользователем команды END MAP содержание ее фиксируется. После этого система анализирует карту, выбирая все различающиеся имена, интерпретирует их в качестве файлов данных и предъявляет пользователю в форме таблиц суперкалька для ввода числовой информации. (Если какие-либо из этих файлов уже созданы в SC4, то система их находит и предъявляет пользователю в текущем состоянии.)

Файлы данных в процессе их разработки можно редактировать без каких-либо ограничений (окончание работы над файлом подтверждается командой END DATA), но и после этой команды к редактированию файла можно вернуться по команде EDIT DATA, после которой система предъявляет пользователю список всех обработанных файлов. В частности, EDIT DATA, предшествующая MPS, позволяет изменить любой элемент данных непосредственно перед формированием MPS-файла, а EDIT MAP – модель. Однако следует помнить, что редактирование карты модели возвращает на начальный этап формирования задачи.

В любой момент работы системы можно ее прервать, а при повторном входе автоматически восстанавливается первоначальная на момент прерыва ситуация. Переход к созданию новой задачи можно сделать на любой стадии. Файлы данных и карты старых задач при этом сохраняются под их именами в файлах *cal*. Уничтожение последних так же, как и выходных файлов системы *mpr*, проводится только самим пользователем. Система поддерживает одновременно несколько пользователей.

Идея генератора заключается в том, что в практических задачах ненулевые элементы распределены по матрице не хаотически, а упакованы в некоторые подструктуры. В результате матрица может быть разбита, причем не единственным способом, на клетки, большая часть которых пуста, часть заполнена элементами типа единичных (псевдоединичных) матриц определенной структуры, описывающих логические связи внутри модели, и лишь небольшая часть – значащей информацией: подматрицами с нормативными данными. Нередко в этих структурах встречается много повторяющихся элементов.

При разработке карты пользователь и создает на экране эту структуру так, как она ему видится, заполняя поле таблицы именами блоков модели в сетке подходящих с его точки зрения координатных имен. Положение каждого блока в карте модели, таким образом, определяется именами строки и столбца, на пересечении которых его разместил пользователь, а в строках (столбцах) не должно быть повторяющихся имен.

1	A	LMGFX	B	P1	E	F	G	H	I	J	K	L	M	N	O
2	X1		E	A	mui	P2	R2	ro2	mu2	P3	R3	ro3	mu3	MAX	
3	V1		-B		E1										
4	Y1		-E2		-E1									E2	
5	M1						E1		E _{am}						
6	X2					E _A	-E1	-1							
7	V2					-B		-m	E1						
8	Y2					-E2								E2	
9	M2								-E1		E1	-1			
10	X3								-B		-E1	-1			
11	Y3								-E2			-m		E2	
13	@											=		<	
14	MIN											=	M3		

Fig. 10

Сами же блоки в случае их идентичности независимо от фактического места в карте должны получать одинаковые имена.

Генератор позволяет вести одновременно и независимо друг от друга несколько задач, реализуя до некоторой степени принцип интегрированной обработки данных, если в картах разных моделей одинаковые элементы данных описываются одним именем, и снабжен системой меню, дающим возможность создавать, уничтожать и сопровождать задачи, редактировать в процессе построения задачи карту модели и таблицы входящих в нее объектов, выводить систему в режимы формирования MPS-файла, обращаться к пакету LP83 для решения задачи с автоматическим формированием командной строки. Отчет пакета с оптимальным решением возвращается в SC4, средствами которого может выполняться любая дальнейшая обработка.

5. ПРИМЕР

На рис. 10 приведена карта модели небольшой динамической задачи, позволяющая проиллюстрировать основные особенности работы с генератором. Введенные в нее имена имеют следующий смысл: E_A – матрица технологических способов производства всех видов продукции и ресурсов, участвующих в межотраслевом обороте; B – матрица коэффициентов приростных фондоемкостей; E_1, E_2 – единичные матрицы, размерности которых соответствуют количествам технологий и продуктов непродовольственного потребления; E_{am} – матрица коэффициентов, определяющих остатки мощностей, переходящие в следующий период; l, m – векторы коэффициентов затрат труда в технологических процессах производства продукции и строительства новых мощностей; M_s, E_{am0} – векторы начальных значений мощностей и их остатков, переходящих во второй период; M_3 – вектор нижних границ конечного состояния мощностей.

Других имен в поле таблицы нет. В оглавлении столбцов, однако, находим имена DMGFX, $P_1, \dots, m_3$, и читатель будет абсолютно прав, если воспримет их как имя задачи и имена блоков ограничений. Аналогично $X_1, \dots, Y_3$ – имена векторов, входящих в набор переменных и, скажем, весь столбец B приведенной таблицы можно прочесть как векторно-матричное уравнение $E_A X_1 - B Y_1 - Y_1 = 0$, а $-X_1 \geq -M_s$ как неравенства в столбце C . Так можно при желании выписать все соотношения модели, приведенной на рис. 10. Символ @ зарезервирован для обозначения строки с описателями типов ограничений, а слова MAX и MIN – для столбца (строки) с именами объектов, определяющими наборы функционалов (правых частей), причем последние два символа можно менять местами в соответствии со смыслом задачи. Отметим еще, что в данном случае мы воспользовались тем, что E_2 – одноэлементная подматрица, единственный элемент которой совпадает с единицей. Знак "–" приводит к умножению всех элементов соответствующей таблицы перед размещением в MPS-файл на -1 , а "+" или пробел все знаки сохраняет. В результате карта модели приобретает естественную форму, близкую к обычной математической символике.

Работа с GLPP2, конечно, требует от пользователя определенных навыков, наиболее трудным из которых является способность описывать интересующую его модель на языке матриц с некоторой внутренней, присущей данному классу задач, структурой клеток, причем чем в большей степени развивается подобное мышление, с тем большей эффективностью используется генератор. Хорошее владение средствами SC4, естественно, тоже не вредит, хотя по наблюдениям за работой аспирантов достаточно свободного владения основными групповыми операциями, чтобы чисто техническая часть работы по заполнению задачи данными становилась довольно простой и не слишком трудоемкой.

В системе обработки данных GLPP2 частично реализован принцип языка APL: структуру объекта можно изменить в любой момент по мере надобности, а одним и тем же именем назвать скаляр, вектор или матрицу. Скаляры представляются как одноэлементные матрицы, векторы – матрицами, содержащими строку или столбец.

Строки (столбцы), не имеющие отличных от нуля элементов, из таблицы исключаются, а в оставшемся блоке заполняются только клетки с ненулевыми значениями элементов. В клетке A1 всегда находится имя объекта, скопированное с карты модели, в строке 1 – имена строк, в столбце A – имена столбцов. Эти имена в дальнейшем при формировании MPS-файла будут присоединяться к координатам объекта в карте модели, образуя полные имена элементов данных. С последними они и войдут в соответствующие структуры MPS-файла, а, значит, полные имена не должны содержать больше 8 символов при известных запретах [8] на использование некоторых символов в именах. Никаких других ограничений на образование имен в GLPP2 нет. Разумеется, действует общее правило программирования: рекомендуется использовать имена с простой и ясной интерпретацией, связанной со смыслом задачи. Таким образом, при фиксированной карте можно произвольно (но согласованно) менять структуру введенных в нее объектов и значения содержащихся в последних числовых данных, получая в ответ оптимальные решения соответствующих задач.

Опыт работы с GLPP2 показал, что скорость лимитируется не ЭВМ, а самим пользователем, точнее, временем, которое ему необходимо, чтобы понять смысл полученных результатов, освоиться с ними и сформулировать следующие интересные с его точки зрения вопросы, гипотезы и т.д. Чисто техническая часть работы, связанная с необходимостью внести изменения в данные и получить результат, составляет лишь 5–20% времени.

При работе в системах типа TRAG, настроенных на определенную модель, пользователь всеми такими возможностями тоже располагает, но в GLPP2 он еще дополнительно может изменить карту, меняя тем самым модель. По опыту работы с унифицированными моделями предполагалось, что эта возможность будет использоваться сравнительно редко. Оказалось, однако, что карта модели меняется, правда, незначительно, а только в некоторых деталях примерно в одном из каждых трех расчетов. Иначе, в ходе расчетов достаточно часто возникают изменения в концепции системы, которые нельзя или трудно выразить на языке одних лишь информационных показателей, меняющих характеристики системы, но не саму систему. Особенно удобно поэтому использовать GLPP2 в начальной стадии, когда концепция системы еще не вполне ясна самим разработчикам. Они получают возможность быстро провести серию расчетов на малоразмерных примерах, моделирующих все качественные особенности будущей большой задачи, и определить достаточно точно как задачу, так и требования к ее информационной базе. Конструирование задачи большой размерности без такой предварительной проработки – весьма опасный эксперимент.

ЛИТЕРАТУРА

1. Канторович Л.В. Математические методы организации производства. Л.: Изд-во ЛГУ, 1938.
2. Канторович Л.В. Экономический расчет наилучшего использования ресурсов. М.: Изд-во АН СССР, 1960.
3. Голдман А.Дж., Таккер А.У. Теория линейного программирования // Линейные неравенства и смежные вопросы. М.: Изд-во иностр. лит., 1959.
4. Гасс С. Линейное программирование. М.: Физматгиз, 1961.
5. Юдин Д.Б., Гольштейн Е.Г. Задачи и методы линейного программирования. М.: Сов. радио, 1964.
6. Малков У.Х. Обзор путей повышения эффективности мультипликативного алгоритма симплекс-метода // Математические методы решения экономических задач. М.: Наука, 1977.
7. Странг Г. Линейная алгебра и ее применения. М.: Мир, 1980.
8. Муртаф Б. Современное линейное программирование. М.: Мир, 1984.
9. Литвин О.Ф. Табличный процессор SUPERCALC4. М.: Финансы и статистика, 1991.

Поступила в редакцию
28 VI 1993