Записки научных
семинаров ПОМИ
Том 312, 2004 г.

И. В. Романовский

О ПРОГРАММИСТСКИХ РАБОТАХ Л. В. КАНТОРОВИЧА С СЕГОДНЯШНЕЙ ТОЧКИ ЗРЕНИЯ

1. Собственно программирование

Сейчас теория и практика программирования ушли чрезвычайно далеко, и кажется, что идеи даже очень крупного ученого, высказанные при самом появлении вычислительных машин, должны безусловно устареть. На самом деле почти так и случилось, но все же кое-что осталось, и полезно на это "кое-что" взглянуть. Нужно начать с рассмотрения той обстановки, в которой Л. В. Канторович заинтересовался вопросами программирования.

В послевоенные годы он возглавлял отдел приближенных вычислений Математического института АН СССР (МИАН). Хотя этот отдел находился в Ленинграде, но вопреки бюрократической очевидности это был отдел именно МИАНа, а не его ленинградского отделения (ЛОМИ). Отдел занимался важными прикладными вычислениями (о его связи с атомным проектом теперь подробно написано в [1]). Основными вычислительными средствами были немецкие настольные машины-полуавтоматы, "Мерседесы" и "Рейнметаллы", чуть ли не трофейные. Расширение вычислительных возможностей было очень важно, и Канторович искал его в разных направлениях.

Одним направлением, которым он очень гордился, было использование для расчетов легкодоступного комплекта счетно-перфорационных машин, закупленного для обработки результатов Всесоюзной переписи 1939 г. и оставшегося после этого без серьезных вычислительных применений. О доступности этой техники Леонид Витальевич узнал еще в 1943 г. на семинаре Л. А. Люстерника в Москве. В 1948 г. Л. В. и Марк Константинович Гаврурин, который был его соавтором не только в довоенной статье по транспортной задаче [2],[1] предложили новые технологии проведения расчетов на этих машинах.[2] Л. В. вспоминал:

> “Основной принцип их эффективного использования – это запараллеливание аналогичных вычислений, благодаря чему появлялась возможность введения простейших программных изменений на коммутационной доске (конечно, вручную). Так, были предложены способы скорой выборки из таблиц и способ расчета скалярного произведения не умножением, а сложением на табуляторе, при этом один из сомножителей формировался не в десятичной, а в двоичной системе [3].
>
> Серьезным конкретным достижением было вычисление на этой примитивной технике таблиц функций Бесселя до 120 порядка на большом интервале [4]. Наиболее интересным тут было запараллеливание вычислений при интегрировании на этих машинах дифференциального уравнения для Бесселевых функций. Запараллеливание достигалось тем, что промежуток интегрирования был разбит на несколько интервалов и функции разных номеров на каждом из интервалов вычислялись одновременно, поэтому получались достаточно большие массивы одинаковых операций, которые эффективно осуществлялись на этих машинах.
>
> Эта работа была сделана при моем участии М. К. Гавуриным и В. Н. Фаддеевой. Любопытно, что параллельно таблицы Бесселевых функций рассчитывались в США на машинах Marc и даже ENIAC. Наша работа началась двумя годами позже и была выполнена всего за полтора года, еще до того, как было закончено издание американских таблиц.[3]”

[1]Эта статья, опубликованная в 1948 г., была написана еще до войны. М. К. Гавурин (1911–1992) был ближайшим учеником и соратником Леонида Витальевича. После отъезда Л. В. в Новосибирск он возглавил кафедру вычислительной математики, а в 1970 г., после отделения он нее двух других кафедр, под давлением администрации перешел (без желания, но к нашей радости) на заведование кафедрой исследования операций.

[2]Машины работали с ужасающим грохотом, и, по-видимому, из-за них Вера Николаевна Фаддеева, одна из ближайших помощниц Л. В. по вычислительным вопросам, всячески противилась впоследствии установке вычислительных машин в здании ЛОМИ.

[3]В только что вышедшей книге [5] описывается история вычислений аналогичных таблиц в конце 40-х годов в США и отмечается, что табли-

Когда появились первые советские ЭВМ, группа Канторовича немедленно начала использовать их для расчетов, хотя машины были доступны только в Москве. Естественно, что о крупных проектах в ближайшем будущем думать не приходилось. Здесь тоже лучше процитировать самого Л. В.:

> "Основным недостатком существовавшей системы, на мой взгляд, было резкое различие языка для машины и математического языка, на котором описывался алгорифм. В математическом описании использовались символы, крупные операции, различные математические понятия, а в машинной программе все это надо было доводить до стандартных операций над числами...
>
> Первым новшеством нашей системы было другое описание вычислительной схемы. Именно, в ней применялись не только арифметические операции, но и многие укрупненные математические операции. Другим, конечно, было то, что элементами схемы были не отдельные числа, а целые их массивы с описаниями расположения и др. В результате схема расчета записывалась с вычислительного плана с логическими связями и переходами, довольно короткого и обозримого.
>
> Для укрупненных операций – упорядочение, скалярное произведение, операции над матрицами и т. д. – были созданы подпрограммы, их реализующие. Также выделялись и специальные операции, если они часто встречались в каком-то расчете. Эта система, в соответствии с модным тогда термином, применявшимся в строительстве, была названа "крупноблочным программированием".
>
> Эти идеи имели и другое воплощение – благодаря использованию логической схемы для записи процесса расчета появлялась возможность оперирования с этими схемами и их применения не только для численных, но и для аналитических выкладок. Скажем, программа позволяла производить аналитическое дифференцирование любой функции, составленной из элементарных или специальных. Эти работы велись начиная с 1953 г., но возможность первых публикаций относится к 1956 г.

цы бесселевых функций требовались при разработке процессов разделения изотопов урана.

> Довести эту работу до конца и создать цельную систему автоматизированного программирования не удалось. Как известно, создание таких систем требует десятков и сотен разработчиков. Кроме того, в нашей системе несколько большее место занимали операции управления, так что при тогдашнем голоде машинного времени ее было целесообразно применять только к массовым задачам. Схемная запись вычислительных планов нашла использование в созданной в Киеве машине “Мир-3”, а в последующем использовалась физиками для проведения на машинах сложных аналитических выкладок.”

Для Л. В. Канторовича и его группы основной была идея “производителя работ”, или, сокращенно, “ПРОРАБа” – еще один распространенный в те времена строительный термин. ПРОРАБом в группе Канторовича называлась специальная программа, которой можно было передать в закодированном виде вычислительное задание, а она бы это задание расшифровала и выполнила. С нашей нынешней точки зрения, это в явном виде интерпретатор. Правда, в отличие от нынешних интерпретаторов сама эта система программирования идейно и технически основывалась на программировании в кодах, так что задание было закодировано в числах. Но нужно иметь в виду, что в то время не было еще ни систем типа АВТОКОДа, ни даже оборудования, позволяющего работать с текстовой информацией.

Группой Канторовича были созданы специализированные ПРОРАБы для задач линейной алгебры. Примечательно, что уже в то время, в начале 1950-х годов, они рассматривали возможность использования компьютеров для проведения символьных вычислений.

Мне бы хотелось еще вернуться к термину “крупноблочное”. Крупноблочным называлось строительство из “укрупненных кирпичей” – тогда впервые стали строить здание не из кирпичей, а из блоков размером, например, в оконную перегородку. Но очень быстро эти блоки стали укрупняться дальше до размеров целой стены, и появился термин “сборочное строительство”, которое существует до сих пор. Вслед за ним не замедлило появиться и “сборочное программирование”.

Собственно, элементы сборочной технологии появились в программировании очень быстро. Начали создаваться *библиотеки* *стандартных программ* и средства поддержки работы с ними. Но это направление очень быстро утратило свою самостоятельность и естественным образом вошло в средства поддержки алгоритмических языков. Когда парк вычислительных машин стал быстро модернизироваться, "стандартные блоки", изготовленные по старинке, т. е. написанные в кодах конкретной машины, моментально погибли, а новые библиотеки были написаны уже на алгоритмических языках.[4]

2. Сборочное программирование в линейном программировании

Сейчас речь пойдет об использовании идей сборочного программирования в одном специальном классе задач линейного программирования.

При решении задач массового раскроя (слово массовый здесь означает, что сырье раскраивается большими партиями, необходимые детали формируются в комплекты и можно не бояться нарушений целочисленности в объеме раскраиваемого сырья) Л. В. Канторович и В. А. Залгаллер [6] предложили замечательный вариант общей схемы – метод, который они называли *методом последовательного улучшения плана* (сейчас мы обычно называем его симплекс-методом, хотя это сильно расширяет принятую трактовку исходного собственно симплекс-метода Дж. Данцига). Идея заключалась в том, что исходная матрица из стандартной задачи линейного программирования заменялась, как мы бы сказали сейчас, "виртуальной". Именно, все предусмотренные методом действия над этой матрицей заменялись соответствующими процедурами, производящими необходимые вычисления способом, удобным для данной конкретной задачи.

[4] И теперь уже возможность их использования фигурирует как требование, ограничивающее развитие систем программирования, – тому примером знаменитая коллекция фортрановских программ. Замечу, что в 50-х годах на семинаре Л. В. Канторовича в ЛОМИ выступал американский специалист, я не помню кто, который говорил о том, что они создают новую систему программирования, название которой будет аббревиатурой от FORmula TRANslator. Л. В. в дискуссии отмечал сильные стороны своего подхода. Американский подход оказался хорош тем, что упростил работу для огромного количества людей и создал возможность гигантской кооперации.

В частности, в задаче линейного программирования при проверке оптимальности текущего опорного плана (допустимого базисного решения) вместо перебора всех мыслимых способов раскроя решалась экстремальная задача, в которой при заданных ценах на выкраиваемые заготовки искался раскрой, обеспечивающий наибольший доход. Значит, если мы пишем программу, реализующую этот метод, то соответствующие участки кода заменяются специальным кодом, который удобно заменить вызовом процедуры – автономно подготавливаемого кода, своего для каждого решаемого класса задач.

Интересно, что эта сугубо программистская идея появилась у авторов, не располагавших программистскими средствами и самостоятельно программ никогда не писавших. Разработанная ими методика счета была очень удобна для ручного счета. Повидимому, первыми программами, реализующими метод генерирования столбцов, были программы, полностью написанные специально для решения задач линейного раскроя [7]. В дальнейшем программы такого типа писали Э. А. Мухачева [8] и я [9]. В моей программе уже достаточно четко выделялись нужные процедуры, и эта структуризация помогла нам легко переносить тот же метод на другие задачи. Впоследствии совершенствование подхода сосредоточилось на улучшении методов решения вспомогательной экстремальной задачи, находящей на каждой итерации раскрой с наибольшей *невязкой*.

Мы разбиваем общую программу симплекс-метода на небольшое число фиксированных функциональных блоков. Эти блоки имеют жестко определенные интерфейсы – правила связи (вызова действий и обмена информацией друг с другом). Каждый из блоков может программироваться в различных вариантах. При выполнении программой конкретного задания она подбирает необходимые блоки (по заданию, введенному из файла или в диалоге с пользователем), составляет из них рабочую программу и исполняет ее. Постепенно мы накопили большой опыт решения важных для приложений задач линейного программирования методами генерирования столбцов.

Необходимо отметить, что интерес к такому подходу очень усилился после того, как Дж. Данциг и Ф. Вольф предложили так называемый декомпозиционный алгоритм для задач линейного программирования блочной структуры [10]. В этих задачах вспомогательная экстремальная задача сама является задачей линейного программирования.

В статье [11] подведен итог наших исследований по разработке методов генерирования столбцов в рамках обычных средств языков программирования. Этих средств было мало для реализации идеи сборочного программирования в алгоритмах симплекс-метода с генерированием столбцов.

В настоящее время мы располагаем такой программой. В ней используются недавно реализованные и ставшие уже очень популярными стандартные конструкции. Я имею в виду предусмотренные в системе MS Windows так называемые DLL – Dynamic Link Libraries (библиотеки динамической связи) [12], которые предназначены именно для сборочного программирования. Необходимая программе DLL может быть вызвана и присоединена к программе прямо в процессе вычислений (так и хочется сказать, как когда-то стандартные программы в интерпретирующей системе ИС-2).

Один из вариантов системы линейного программирования, основанный на применении DLL, был представлен в недавно защищенной диссертации моей аспирантки Ли Херан [13]. Подробности будут опубликованы в статье И. В. Романовского, С. С. Сурина и Ли Херан в Вестнике СПбГУ.

Литература

1. В. Л. Канторович, С. С. Кутателадзе, Я. И. Фет, *Леонид Витальевич Канторович: человек и ученый*, т. 1. Изд-во СО АН, Новосибирск (2002).
2. Л. В. Канторович, М. К. Гавурин, *Применение математических методов в вопросах анализа грузопотоков*. — В сб.: Проблемы повышения эффективности работы транспорта. Изд. АН СССР, М.-Л. (1949), сс. 110–138.
3. Л. В. Канторович, М. К. Гавурин, *О некоторых новых приемах вычислений на табуляторе, связанных с использованием двоичных разложений чисел*. — Усп. мат. наук **3**, вып. 4 (1948), 160–162.
4. М. К. Гавурин, В. Н. Фаддеева, *Таблицы функций Бесселя $J_n(x)$ целых номеров от* 0 *до* 120. Гостехиздат, М.-Л. (1950).
5. N. Campbell-Kelly, N. Croaken, R. Flood, E. Robson (eds), *The History of Mathematical Tables. From Sumer to Spreadsheets.* Oxford University Press, Oxford (2003).
6. Л. В. Канторович, В. А. Залгаллер, *Расчет рационального раскроя промышленных материалов*. Лениздат, Ленинград (1951).
7. В. А. Булавский, М. А. Яковлева, *О решении задачи оптимального раскроя линейных материалов на ЭВМ*. — В сб.: Линейное программирова-

ние (Тр. науч. совещ. о применении матем. методов в эконом. исслед. и планировании) **4**, М. (1960).
8. Э. А. Мухачева, *Рациональный раскрой прямоугольных листов на прямоугольные заготовки.* — Оптимальное планирование, вып. 6, Новосибирск (1966).
9. И. В. Романовский, *Программа решения задачи линейного раскроя.* — Оптимальное планирование, вып. 12, Новосибирск (1969).
10. G. Dantzig, Ph. Wolfe, *Decomposition principle for linear programs.* — Operations Research **8**, No. 1 (1960), 101–110.
11. Л. В. Канторович, И. В. Романовский, *Генерирование столбцов в симплекс-методе.* — Экономика и математические методы **21** (1985), 128–138.
12. Д. Соломон, М. Русинович, *Внутреннее устройство Microsoft Windows 2000. Мастер-класс.* Питер, СПб (2001).
13. Ли Херан, *Генерирование столбцов в симплекс-методе (вопросы программной реализации).* Диссертация на соискание ученой степени кандидата физико-математических наук, СПб (2002).

Romanovsky J. V. A nowadays view on Kantorovich's works in software.

L. V. Kantorovich dealt with computers when they only appeared in the Soviet Union. That time he headed a serious computational project and searched for better computational tools. First we describe his attempts to improve the computing routines and to propose the idea of interpretation of a computational plan. Second we consider the fate of the fruitful idea of column generation in linear programming.

С.-Петербургский государственный университет
E-mail: jvr@JR2793.spb.edu

Поступило 25 августа 2004 г.