Л. Г. Хачиян

Избранные труды

МЦНМО

УДК 510.52+519.85
ББК 22.18
Х29

Хачиян Л. Г.
Избранные труды
Электронное издание
М.: МЦНМО, 2014
519 с.
ISBN 978-5-4439-2089-4

В сборник избранных трудов Л. Г. Хачияна (1952–2005) вошли наиболее значительные работы по сложности задач линейного и математического программирования, а также по теории дуализации и генерации. Подробно, в нескольких авторских вариантах, изложен открытый Л. Г. Хачияном полиномиальный алгоритм решения задачи линейного программирования — фундаментальный вклад Л. Г. Хачияна в математическое программирование.

Книга будет полезна специалистам в области математического программирования и теории сложности, аспирантам и студентам.

Подготовлено на основе книги: *Л. Г. Хачиян*. Избранные труды. — М.: МЦНМО, 2009.

Издательство Московского центра
непрерывного математического образования
119002, Москва, Большой Власьевский пер., 11,
тел. (499) 241-74-83.
`http://www.mccme.ru`

ISBN 978-5-4439-2089-4

Содержание

От составителя 5
А. С. Немировский. Вместо предисловия 12
Биографическая справка 24

Раздел I. АЛГОРИТМЫ ДЛЯ МАТРИЧНЫХ ИГР 25

О скорости сходимости игровых процессов решения матричных игр 26
Сублинейный приближенный вероятностный алгоритм для матричных игр (*совм. с М. Григориадисом*) 38

Раздел II. СЛОЖНОСТЬ ВЫПУКЛОГО ПРОГРАММИРОВАНИЯ 49

Сложность выпуклых задач вещественного и целочисленного полиномиального программирования 50
Полиномиальный алгоритм в линейном программировании 202
Авторское доказательство полиномиальности линейного программирования 207
Полиномиальные алгоритмы в линейном программировании 224
О точном решении систем линейных неравенств и задач линейного программирования 244
Полиномиальная разрешимость выпуклого квадратичного программирования (*совм. с М. К. Козловым и С. П. Тарасовым*) 249
Метод вписанных эллипсоидов (*совм. с С. П. Тарасовым и И. И. Эрлихом*) 253
Диагональное шкалирование матриц и линейное программирование (*совм. с Б. Калантари*) 260
Диагональное шкалирование матриц NP-трудно 267
Сложность полуопределенного программирования (*совм. с Л. Порколабом*) 273
Целые точки в выпуклых полуалгебраических множествах (*совм. с Л. Порколабом*) 289
Округление многогранников в вещественной модели вычислений 311

Раздел III. ТЕОРИЯ ДУАЛИЗАЦИИ И ИНКРЕМЕНТАЛЬНОГО ПОРОЖДЕНИЯ 329

В. А. Гурвич. Эффективность алгоритмов и сложность задач перечисления 330

О сложности дуализации монотонных дизъюнктивных нормальных форм (*совм. с М. Фредманом*) 356

О генерации тупиковых конъюнктивных и дизъюнктивных нормальных форм монотонных булевых функций (*совм. с В. А. Гурвичем*) 367

Двойственно-ограниченные задачи генерации: порождение минимальных целочисленных решений монотонной системы линейных неравенств (*совм. с Е. Борошем, В. А. Гурвичем и др.*) 379

Раздел IV. РАЗНОЕ 407

Применение псевдополиномиальных алгоритмов для некоторых задач комбинаторной оптимизации с ограничениями (*совм. с Ю. Г. Сметаниным*) 408

Задача вычисления объема многогранника перечислительно трудна 415

Проводимость цепи Маркова на частичном порядке (*совм. с А. В. Карзановым*) 418

Циклические игры и нахождение минимаксных средних циклов в ориентированных графах (*совм. с В. А. Гурвичем и А. В. Карзановым*) 427

Раздел V. MEMORIA 441

В. К. Леонтьев. Чтобы помнили... 442

В. Хватал. Вспоминая Леонида Хачияна 444

М. К. Козлов. Особенности национальной науки в эпоху застоя 446

Х. Эльбассиони Две лекции Леонида Хачияна 453

В. А. Гурвич. Леонид Хачиян и все-все-все 462

Публикации Л. Г. Хачияна 504

Именной указатель 513

Вместо предисловия

А. С. Немировский

Приглашение участвовать в комментировании работ Леонида Генриховича Хачияна заставило меня задуматься: что, собственно, комментирование в данном случае означает? То, что Леонид успел сказать за не так уж щедро отпущенные ему судьбой годы, он сказал сам, причем в безукоризненно ясной форме, и никакие комментарии тут не требуются. Описывать «человеческую компоненту» его исследований — так сказать, «Пушкин в жизни» — не по моим силам по многим причинам, например по скудости личных контактов. Единственный вид комментария, который мог бы быть интересен для читателя и посилен для меня, — это рассказ о последствиях одного из самых замечательных и уж точно самого знаменитого из результатов Леонида (доказательство полиномиальной разрешимости линейного программирования) для одной из важных областей прикладной математики — алгоритмов оптимизации. Поскольку и я сам работаю в этой области, последующее можно считать свидетельством очевидца, со всеми вытекающими из этого потенциальными плюсами (прямое, а не понаслышке знание дела) и минусами («врет как очевидец»). Читатель должен иметь в виду, что последующее изложение есть хотя и профессиональное, но все же личное мнение; другие специалисты могут видеть вещи по-другому...

Согласно «Толковому словарю живого великорусского языка В. И. Даля», математика есть «наука о величинах и количествах; все, что можно выразить цифрою, принадлежит математике. — *чистая*, занимается величинами отвлеченно; — *прикладная*, прилагает первую к делу, к предметам». По этому (да и по любому другому) определению Леонид Хачиян, несомненно, «чистый» математик; и тем не менее (по крайней мере) один из его результатов, сам по себе достаточно отвлеченный, весьма сильно повлиял на приложения математики «к делу, к предметам». Цель настоящей заметки — объяснить, как и почему это произошло; но начать следует с приложений, о которых идет речь.

Математическое программирование (МП). Эта область прикладной математики занимается качественным исследованием и численным решением задач вида «максимизировать заданную функцию $f(x)$ от вектора переменных $x = (x_1, \ldots, x_n)$ при ограничениях $g_i(x) \leqslant b_i$, $1 \leqslant i \leqslant m$»;

таковы математические модели «жизненных» задач принятия решений в ситуациях, когда само решение описывается конечным набором чисел x (скажем, времен, отводимых в производственном плане каждому из заданного числа n возможных производственных процессов), а его последствия характеризуются набором численных критериев (результирующий выход продуктов, потребные ресурсы, расходы, прибыль и т. п.), представляющих собой известные функции от x. В ситуации такого рода содержательные задачи типа «выбрать решение, оптимизирующее один из критериев при ограничениях на остальные», например, найти производственный план, максимизирующий прибыль при заданных верхних границах на ресурсы и нижних границах на выпуск продуктов, естественно моделируются как задачи МП.

Линейное программирование (ЛП) — это специальный случай МП (исторически МП из него и выросло), в котором и целевой функционал $f(x)$, и ограничения $g_i(x)$ — линейные функции x. Как математический объект ЛП обладает вполне прозрачной структурой, позволяющей развить элегантную и исчерпывающую теорию (содержательная интерпретация которой, помимо всего прочего, во многом объясняет механизм образования цен). С прикладной точки зрения несомненным достоинством ЛП является относительная простота описания модели (задать линейную функцию куда проще, чем нелинейную, что существенно упрощает сбор необходимых для построения модели данных) и в особенности наличие, притом с самого рождения ЛП, чрезвычайно эффективных (если и не в теории, то уж точно на практике) алгоритмов численного решения задач ЛП. Эта тема — эффективные алгоритмы решения задач ЛП — и есть предмет нашего рассказа.

Начнем с того, что чистая математика — та, которая, по Далю, «занимается величинами отвлеченно»,— наука в основном дескриптивная, нацеленная прежде всего на выяснение структуры и свойств математических объектов и связей между этими объектами; главная цель здесь — понять, как устроен «математический мир» (что, как хорошо известно, есть один из основных ключей к пониманию того, как устроен мир реальный). Операциональные аспекты — как перейти от «косвенного» описания объекта (скажем, уравнения, описывающего нечто) к его «явному» описанию (численному представлению этого нечто) — для чистой математики обычно имеют второстепенный интерес: в самом деле, явное численное описание сплошь и рядом мало что добавляет к пониманию устройства «математического мира». Иное в математике прикладной — здесь конечной целью обычно является «получение числа», и операциональные аспекты становятся центральными. К примеру, «чистый» математик, занимаясь задачей МП, постарается выяснить, есть ли у нее решение, единственное ли оно и как его охарактеризовать (каковы необходимые и/или достаточные условия оптимальности). Все эти вопросы важны и для прикладного математика, но для него они далеко не исчерпывают сути дела: его конечной целью является построение вычислительного алгоритма, который позволяет найти решение, точное или приближенное заданной точности, в явной численной форме — без этого решение не годится для практического употребления[1]). В случае ЛП операциональная компонента присутствовала с самого начала — это *симплекс-метод*, придуманный основателем ЛП[2]) Дж. Данцигом) (G. Dantzig) «в комплекте» с самим понятием задачи линейного программирования (1947). Этот алгоритм оказался чрезвычайно успешным — сначала при решении «на руках» совсем маленьких ЛП, а по мере появления и совершенствования компьютеров (и чисто алгоритмических усовершенствований самого метода) — при решении все бо́льших и бо́льших задач ЛП, вплоть до десятков и сотен тысяч (а в отдельных случаях — и миллионов) переменных и ограничений. Этот процесс сопровождается постоянным расширением круга практических приложений ЛП, в том числе для оптимизации: кормовых смесей в птицеводстве и животноводстве, использования сырья в химических производствах, управления запасами, планирования перевозок и других материальных потоков в сетях, планирования производства, составления расписаний, планирования сеансов радиационной терапии — этот перечень можно продолжать и продолжать[3]).

---

[1]) В этой связи совершенно естественно, что становление и развитие МП/ЛП четко скоррелировано с появлением и развитием компьютеров (при том, что «чистая математика» занимается оптимизационными задачами едва ли не с самого своего возникновения).

[2]) Уместно заметить, что концепция задачи ЛП как основы для принятия оптимальных решений была предложена блестящим советским математиком Л. В. Канторовичем еще в 1939 г. в брошюре «Математические методы организации и планирования производства» (ЛГУ); там же предложено и нечто вроде алгоритма («метод разрешающих множителей»). Работа Л. В. Канторовича не была замечена на Западе (и в этом смысле никак не повлияла на становление ЛП) и не получила одобрения в СССР (какая же может быть математика, если речь идет об экономике, да еще марксистско-ленинской!). Л. В. Канторович потратил огромное количество усилий и нервов в попытках доказать советским ортодоксам, что ЛП и прочие «математические методы в экономике», над которыми он работал до конца жизни, не противоречат марксизму, по тому времени скорее «верному, потому что всесильному», чем «всесильному, потому что верному». Как часто случалось в тот период, настоящее признание пришло к Л. В. Канторовичу с Запада — в виде (разделенной с Т. Купмансом) Нобелевской премии по экономике «за вклад в теорию оптимального распределения ресурсов» (1977).

[3]) Дополнительным и, возможно, самым важным «клиентом» алгоритмов ЛП являются методы решения задач дискретной оптимизации — оптимизационных задач, в которых отдельные или все переменные должны принимать только целые значения или только значения {0, 1}. Связывая «дискретные» и «непрерывные» переменные линейными неравенствами, можно моделировать практически любые ограничения на принимаемые решения, как чисто количественные, так и логико-структурные. Задачи дискретной оптимизации имеют огромное поле приложений и, как правило, чрезвычайно трудны. Практически все существующие вычислительные методы для дискретных задач интенсивно используют алгоритмы ЛП и предъявляют весьма жесткие требования к производительности последних.

Симплекс-метод: теория и практика. Для нашего рассказа полезно дать читателю представление о том, как работает симплекс-метод. Для понимания этого достаточно вообразить себе выпуклый многогранник в трехмерном пространстве — это будет множество допустимых (т. е. удовлетворяющих ограничениям) точек. Наша цель — найти в допустимом многограннике «самую высокую точку» — точку с наибольшей z-координатой (z, стало быть, наш целевой функционал). Искомая точка будет, конечно, одной из вершин многогранника, и чтобы достичь ее, симплекс-метод делает следующее. Находясь в некоторой вершине, мы проверяем, исходит ли из нее такое ребро, двигаясь по которому, мы будем подниматься (z-координата будет расти). Если такого ребра нет, дело сделано — мы уже находимся в самой высокой вершине многогранника (или в одной из таких вершин, если их много). Если «поднимающее ребро» есть, дойдем по нему до его конца, оказавшись, таким образом, в новой вершине многогранника, лежащей выше той вершины, которую мы покинули. В этой новой вершине мы повторим те же действия, в результате которых либо убедимся, что цель уже достигнута, либо поднимемся еще выше, и т. п.Через конечное число шагов мы и в самом деле достигнем самой высокой вершины — ведь всего вершин конечное число, а в нашем движении мы только поднимаемся и потому не можем посетить одну и ту же вершину дважды. Приведенное интуитивно-геометрическое описание может быть переведено на алгебраический язык и распространено со случая трехмерного многогранника (отвечающего ЛП с тремя переменными) на общий случай, но суть дела остается прежней: получив «на входе» данные задачи ЛП (коэффициенты целевого функционала и ограничений и правые части этих ограничений), симплекс-метод решает задачу[1]) за конечное число арифметических операций над этими данными (предполагается, что операции выполняются точно).

С классической точки зрения симплекс-метод — алгоритм идеальный: что может быть лучше, чем гарантированно конечный алгоритм, ничего сверх обычной арифметики не использующий! Следует, однако, принять во внимание, что четкая дихотомия «конечное — бесконечное» имеет в основном чисто дескриптивный смысл; с операциональной точки зрения (попросту говоря, для простых смертных, которым приходится ждать ответа от компьютера) «конечное, но очень большое» ничем не отличается от бесконечного. «Чудо» симплекс-метода состоит в том,

---

[1]) То есть либо строит одно из оптимальных решений, либо приходит к правильному заключению, что таковых нет ввиду противоречивости системы ограничений, либо, наконец, приходит к правильному заключению, что оптимальных решений нет в силу того, что целевой функционал не ограничен сверху на множестве допустимых решений (есть бесконечное ребро, уходящее «вверх»); никаких возможностей, помимо трех указанных, в ЛП не существует.

что, как показывает почти 60-летний вычислительный опыт, требуемое им фактическое число операций не просто конечно, но и не очень велико — на практике симплекс-метод обычно решает задачу, посетив всего $O(1)(m+n)$ вершин допустимого многогранника (m — число ограничений, n — число переменных) с вполне разумной константой $O(1)$, что-то вроде четырех. В результате метод способен решать за разумное время достаточно большие задачи, что, надо думать, во многом определило практический успех ЛП, а косвенно — и интенсивное развитие этой и смежных областей прикладной математики, в первую очередь — МП и дискретной оптимизации. Поразительная практическая эффективность симплекс-метода названа выше «чудом» отнюдь не случайно: в самом устройстве метода нет ничего, что объясняло бы этот феномен, скорее напротив. В самом деле, единственное, что можно твердо гарантировать — это то, что число шагов, т. е. число посещенных в процессе решения вершин допустимого многогранника, не превзойдет общего числа вершин этого многогранника; по нашим сведениям, существенно лучших гарантированных оценок сверху на число шагов нет и по сей день. Ну а число вершин многогранника, заданного m линейными ограничениями на n переменных, может быть (и обычно бывает) астрономически большим уже при вполне умеренных m и n. К примеру, обычный n-мерный куб $\{x=(x_1, \ldots, x_n) \mid -1 \leqslant x_i \leqslant 1,\ i=1, \ldots, n\}$ задается $m=2n$ линейными неравенствами на n переменных и имеет 2^n вершин (что больше 10^{30} уже при $n=100$); если бы симплекс-метод, работая на кубе, должен был бы и в самом деле перебрать сколько-нибудь заметную долю от общего числа вершин, не могло бы быть и речи о решении этим методом общих задач ЛП уже с немногими десятками ограничений и переменных. На самом деле легко доказать, что, работая в точности на n-мерном кубе, симплекс-метод приходит в оптимальную вершину не более чем за n шагов. В то же время знаменитый пример В. Кли и Д. Минти (V. Klee, G. J. Minty, 1972) показывает, что слегка «возмутив» куб, можно действительно «заставить» симплекс-метод посетить экспоненциально много вершин. В этом примере допустимый многогранник — это множество

$$\begin{gathered} 0 \leqslant x_1 \leqslant 1, \\ \varepsilon x_1 \leqslant x_2 \leqslant 1-\varepsilon x_1, \\ \ldots\ldots\ldots\ldots \\ \varepsilon x_{n-1} \leqslant x_n \leqslant 1-\varepsilon x_{n-1}, \end{gathered}$$

где ε — параметр. Легко убедиться, что когда ε положительно и меньше $1/2$, то наш многогранник с топологической точки зрения устроен совершенно так же, как и куб, в частности, задается $m=2n$ линейными неравенствами на n переменных и имеет 2^n вершин. Теперь, однако, нетрудно указать маршрут, проходящий через все 2^n вершин, начиная от «самой нижней» (с наименьшим возможным значением x_n) и кончая «самой верхней» (с наибольшим возможным значением x_n), и согласованный с правилами симплекс-метода (из текущей вершины в следующую мы переходим не произвольно, а по ребру, и при этом поднимаемся — x_n растет)[1]). В свете такого рода примеров удивительная практическая эффективность симплекс-метода действительно представляется чудом (лишь отчасти объясняемым имеющимися результатами о хорошем поведении метода «в среднем»).

Проблема оценки сложности ЛП. Прикладная математика, хоть она и прикладная, все же математика, и многие прикладные математики ищут «доказуемо хорошие» вычислительные алгоритмы, причем доказуемость допускается только «первой свежести» — формально-математическая; вычислительная практика, как бы обширна и убедительна (в человеческом, а не формальном понимании этого слова) она ни была, здесь в зачет не идет. Разумеется, нельзя доказать формально, что нечто хорошо, не имея формального же определения того, «что такое хорошо и что такое плохо». По этому последнему поводу в середине 1960-х годов благодаря усилиям и весьма нетривиальным результатам многих замечательных математиков выработалось определенное согласие, сохраняющееся уже почти 50 лет и, по всей видимости, вряд ли подлежащее пересмотру в обозримом будущем. Применительно к нашей теме — вычислительным алгоритмам ЛП — соответствующая концепция «хорошести» (эффективности) вычислительного алгоритма ведет к двум возможным определениям (в которых я опускаю неуместные здесь сугубо технические детали).

А. Эффективный вычислительный алгоритм для ЛП — это программа для идеализированного «вещественно-арифметического» компьютера (умеющего хранить вещественные числа и совершать точные арифметические операции над ними), которая, получив на входе натуральные

[1]) Читатель может удивиться: как же так — «практика демонстрирует высокую эффективность», а совсем простые, явно выписываемые примеры показывают возможность полной катастрофы. Почему бы не взять и не напустить симплекс-метод на такой пример и не объявить это частью практики? Дело, однако, обстоит не так просто. Как и многие другие методы вычислительной математики, симплекс-метод — это лишь частично определенный алгоритм: какие-то элементы конструкции фиксированы, а остальные могут варьироваться в определенных пределах. Например, двигаться по ребрам и подниматься — эта часть симплексной стратегии фиксирована, а вот как выбирать «поднимающее ребро», когда их несколько, — этот элемент конструкции не фиксирован и меняется от реализации к реализации. В результате реальные версии симплекс-метода не обязаны идти — и де-факто не идут — по самому длинному маршруту, благо помимо него в указанном примере «самая нижняя» и «самая верхняя» вершины непосредственно связаны ребром, выбор которого решает задачу за 1 шаг, и столь же совместим со стратегией симплекс-метода.

числа m, n и набор $m(n+1)+n$ вещественных чисел — коэффициентов целевого функционала и ограничений задачи ЛП с n переменными и m ограничениями — за *полиномиальное по m, n число арифметических операций* находит точное решение задачи (т. е. либо выдает оптимальные значения переменных, либо выдает корректное заключение о том, что задача неразрешима). Указанное должно иметь место при всех m, n и для всех возможных наборов вещественных коэффициентов.

Б. Эффективный вычислительный алгоритм для ЛП — это программа для «рационально-арифметического» компьютера, имеющего ячейки фиксированной длины и умеющего выполнять арифметические и логические операции с содержимым этих ячеек, которая обрабатывает фиксированное число ячеек за один такт. Получив на вход двоичное слово длины L, кодирующее натуральные числа m, n и набор $m(n+1)+n$ *рациональных* чисел — коэффициентов целевого функционала и ограничений задачи ЛП с n переменными и m ограничениями и *рациональными данными* — программа за *полиномиальное по L* время находит точное решение задачи (последнее понимается в том же смысле, что и в версии А). Указанное должно иметь место при всех m, n и для всех возможных наборов рациональных коэффициентов.

В связи с версией Б отметим, что среди решений разрешимой задачи ЛП с рациональными данными всегда есть рациональное, и рационально-арифметический компьютер вполне в состоянии его выдать.

С чисто математической точки зрения центральный вопрос, связанный с той или иной вычислительной задачей, это вопрос о ее «сложностном статусе». Применительно к ЛП, этот вопрос звучит так:

Допускает ли ЛП эффективный разрешающий алгоритм? (?)

На самом деле здесь не один вопрос, а два, соответственно двум интерпретациям — А и Б, понятия эффективности. Следует подчеркнуть, что вопросы эти существенно разные. В версии А мы разрешаем компьютеру очень многое: хранить «настоящие» вещественные числа (т. е. бесконечные последовательности битов) и точно оперировать с ними, считая при этом, что выполнение одной арифметической операции с вещественными операндами «стоит» единицу времени; это, разумеется, весьма сильная идеализация. Но мы и многого хотим: задача ЛП должна решаться за время, полиномиальное по ее размерам (числу переменных и числу ограничений) и никак не зависящее от значений коэффициентов, которым разрешается быть любыми вещественными числами. В версии Б мы нисколько не идеализируем компьютер — теперь и речи нет о вещественных числах и точной вещественной арифметике; все, что мы теперь можем, это оперировать с числами рациональными, имея при этом в виду, что длительность элементарной операции над такими числами (скажем, сложения) уже не занимает единицу, как было в модели точной вещественной арифметики, а эта длительность тем больше, чем больше разрядность складываемых чисел. Но мы и меньше требуем: теперь «хорошему» алгоритму разрешается долго решать даже совсем маленькую по размерам задачу (скажем, одна переменная и одно ограничение), если только данные этой задачи — достаточно длинные рациональные числа. В самом деле, теперь время работы эффективного алгоритма должно быть ограничено полиномом от длины входа (общего числа битов в записи всех коэффициентов), а не от числа коэффициентов (которое, разумеется, никогда не бывает больше, чем длина входа, и может быть на сколько угодно порядков меньше этой длины).

Итак, перед нами две существенно разных версии вопроса (?). Как насчет ответов?

В интерпретации А вопрос об эффективной разрешимости ЛП открыт и по сей день; он числится под № 9 в составленном в 1998 г. С. Смейлом (S. Smale, Филдсовский медалист 1966 г.) списке из 18 открытых проблем для математики XXI века (аналог знаменитого списка из 23 проблем, поставленных Давидом Гильбертом в 1900 г. перед математикой века XX).

В интерпретации Б вопрос об эффективной разрешимости ЛП был положительно решен Леонидом Хачияном в 1978 г. Об этом и рассказ.

Результат. По существу Леонид получил свой результат, «скрестив» постановки А и Б. А именно, он использовал чрезвычайно простой алгоритм[1]) *приближенного* решения задач так называемого выпуклого программирования (включающего ЛП с вещественными коэффициентами) на *вещественно-арифметическом* компьютере для *точного* решения задач ЛП с *рациональными* коэффициентами на *рационально-арифметическом* компьютере. Из теории метода эллипсоидов непосредственно вытекает, что *если в задаче ЛП с n переменными и m ограничениями последние включают явные границы $|x_i| \leqslant R$, $i = 1, \ldots, n$ на все переменные, а величины всех коэффициентов не превосходят некоторого M, то при любом $\varepsilon > 0$ задачу можно решить с точностью ε* (т. е. либо корректно заключить, что задача несовместна, либо выдать ее приближенное решение, в котором ограничения нарушаются разве что на ε и значение целевого функционала разве что на ε хуже оптимального) *за полиномиальное по n, m* и $\log(RM/\varepsilon)$ *число операций точной вещественной арифметики*. Открытие Леонида состояло в том, что уже этого,

---

[1]) Метод эллипсоидов предложен в 1976 г. Д. Б. Юдиным (1917—2006) и автором этих строк и независимо, чуть позже — Н. З. Шором (1937—2006).

на самом деле чрезвычайно простого факта (бог весть почему оставшегося необнаруженным за 25 лет развития МП) достаточно, чтобы ответить положительно на «рационально-арифметическую» версию вопроса (?). А именно, Леонид показал, что *точное* решение задачи ЛП с *рациональными* данными и длиной входа L сводится к приближенному решению этой же задачи с точностью $\varepsilon = 2^{-O(1)\,L}$, и притом с добавленными границами на переменные $|x_i| \leqslant 2^{-O(1)\,L}$. С помощью метода эллипсоидов приближенное решение нужной точности можно найти за полиномиальное по n, m и и $\log(RM/\varepsilon)$ (а тем самым, в силу совсем простых оценок, полиномиальное по L) число операций точной вещественной арифметики. Последний шаг в конструкции Леонида (технически неприятный, но по сути дела простой) состоял в демонстрации того, что требуемые операции точной вещественной арифметики можно выполнять приближенно, удерживая $O(nL)$ знаков до и после запятой. Ну а такая работа посильна и для рационально-арифметического компьютера; и поскольку как общее число требуемых арифметических операций, так и число знаков, которые в них надо удерживать, полиномиальны по L, таковым же оказывается и общее число тактов работы этого компьютера.

Краткосрочные последствия. Результат Хачияна вызвал настоящую сенсацию на Западе. Специалисты по оптимизации были «взбудоражены» решением фундаментальной для них и долгое время открытой проблемы и интенсивно переживали это событие, что выразилось в настоящем «извержении» профессиональных публикаций по предмету; это совершенно естественно и ничуть не удивительно. Что действительно удивительно, это сенсация, вызванная открытием Хачияна, в том, что теперь (не помню, как тогда) называется массмедиа, в том числе и в центральных газетах США (включая New York Times[1])), Европы и Японии. При этом авторы статей в непрофессиональной прессе не слишком старались, да и при всем желании не могли правильно передать содержание результата и создали впечатление, что речь идет не об академическом достижении, а о вполне реальном прорыве в возможностях решать практические задачи принятия решений, включая и наиболее трудные из них — задачи дискретной оптимизации. О подробностях всей этой истории можно прочитать в статье Ю. Лоулера (E. Lawler) с характерным названием The Great Mathematical Sputnik of 1979 (Science, September 1979).

Наряду с непрофессиональными (и скоро стихшими) восторгами и множеством быстро написанных профессиональных статей по поводу результата Хачияна и метода эллипсоидов (мало что прибавивших к тому, что с самого начала было сделано в России), нашлись — и это,

---

[1]) 11 ноября 1979 г. По-видимому, это первый в истории случай, когда чисто математическое событие удостоилось такого внимания прессы.

пожалуй, наиболее удивительно — профессионалы, которые стали проверять, действительно ли метод эллипсоидов в реальных вычислениях может конкурировать с симплекс-методом. Любому нормальному человеку ответ был ясен заранее: метод эллипсоидов при всей своей теоретической эффективности явно не в состоянии справляться за разумное время с задачами ЛП с всего-навсего двумя-тремя десятками переменных (что, между прочим, никак не противоречит его теоретической оценке трудоемкости)[1]). Ну а симплекс-метод, сколь бы плохо ни было его поведение в наихудшем случае, на практике прекрасно решает задачи с многими тысячами и десятками тысяч переменных. Как и следовало ожидать, эксперименты обернулись для метода эллипсоидов полной катастрофой, а для симплекс-метода — полным триумфом, в полном соответствии с замечательной русской пословицей «правда хорошо, а счастье лучше». Так что если говорить о практике ЛП и МП, сенсация кончилась, и все вернулось на круги своя.

Долгосрочные последствия. Начну с замечательного анекдота. Путешественники на воздушном шаре потеряли ориентировку; вдруг порывом ветра шар приблизило к земле, они видят на ней человека и кричат ему: «Где мы?» После паузы слышен ответ: «Вы на воздушном шаре». Новый порыв ветра уносит шар в облака, где у путешественников происходит следующий диалог: «Ну ясно, это был математик. — Откуда вы знаете? — Во-первых, он подумал, прежде чем отвечать. Во-вторых, ответ был совершенно точен. А в-третьих, он был абсолютно бесполезен». Из того, что рассказано к настоящему моменту, легко сделать вывод, что по своим практическим последствиям обсуждаемый результат Леонида вполне согласуется с процитированным анекдотом. Я, однако, полагаю, что дело обстоит далеко не так, и сейчас постараюсь убедить в этом читателя. Результат Хачияна не оказал немедленного и прямого воздействия на вычислительную практику ЛП, это верно; но он оказал немедленное и прямое воздействие на исследователей, которые при этой практике состоят. Этим исследователям была предъявлена альтернатива симплекс-методу, обладающая явным, хотя и чисто теоретическим преимуществом, и тем самым наглядно продемонстрировано, что этот привычный и замечательный алгоритм никоим образом не Тора с Синая[2]). Я думаю, не будет большим преувеличением сказать,

[1]) При всей своей теоретической значимости метод эллипсоидов как средство реальных вычислений имеет достаточно скромную, хотя и не пустую, область применения, которая не имеет ничего общего с ЛП. И сегодня, при выросшей более чем на три порядка производительности компьютеров, этот метод в состоянии решать задачи разве что с сотней, максимум парой сотен переменных (число ограничений особой роли не играет).

[2]) На самом деле альтернативы симплекс-методу предлагались уже при рождении ЛП и притом такими незаурядными людьми, как Дж. фон Нейман и Т. Моцкин; однако

что к середине 1970-х годов у подавляющего большинства западных оптимизаторов сама идея замены симлекс-метода чем-то существенно новым психологически была прочно заблокирована. Это в СССР оптимизаторы могли «думать о божественном», не особенно обращаясь к компьютеру, потому что и социального заказа на оптимизацию почти не было, и компьютеры были из рук вон плохи; ну а на Западе, при волчьих законах капитализма, прикладным математикам, даже университетским преподавателям с их пожизненно гарантируемыми должностями, в массе куда как менее естественно напрочь игнорировать вычислительную практику. Отсюда, я думаю, и психологические блоки... А тут вдруг прямое, хоть и неудавшееся, покушение на «священную корову». Полагаю, что психологический эффект от полученного Леонидом результата (и вызванной этим результатом шумихи) был таков, что люди начали думать о том, о чем раньше не думали. И результаты не заставили себя долго ждать: уже в 1984 г. молодой (как и можно было ожидать) исследователь, Нарендра Кармаркар, в то время начинающий сотрудник Bell Laboratories, объявил об открытии нового алгоритма ЛП, который не только теоретически эффективен, но и на практике способен конкурировать с симплекс-методом[1]). Описание возникшей сенсации и дальнейшего развития событий выходит за рамки моего рассказа; скажу только, что напрямую инициированный открытием Кармаркара период чрезвычайно бурного развития ЛП и МП, обычно называемый Interior Point Revolution[2]), осуществлялся совокупными усилиями многих ученых и завершился лишь к концу 1990-х годов. Многие думают, что по своей плодотворности этот период сопоставим с «баснословными» временами становления и бурного первоначального развития ЛП и МП. Так ли это, покажет будущее, я же ограничусь простым перечислением некоторых относящихся к этой революции фактов.

• Было открыто семейство совершенно новых *теоретически эффективных алгоритмов ЛП — методов внутренней точки*, которые были затем распространены с сохранением эффективности на широкие классы «хорошо устроенных» (а именно выпуклых) существенно нелинейных оптимизационных задач. Методы внутренней точки охвачены единой глубокой

результаты численных экспериментов были явно в пользу симплекс-метода, и об альтернативах быстро забыли — сколь-нибудь заметных теоретических преимуществ перед симплекс-методом у них не было, а считали они куда хуже.

[1]) Хотелось бы отметить, что в конце 1960-х годов иркутский математик И. И. Дикин (1936—2008) едва ли не первым предложил использовать для решения ЛП методы внутренней точки. — *Примеч. сост.*

[2]) Я бы и рад перевести это название на русский, да выходит как-то смешно — что это еще за «революция внутренней точки»? На самом деле имеется в виду, что новые методы решения задач ЛП движутся по внутренним точкам допустимого многогранника и тем отличаются от симплекс-метода. Сколь бы неудачным ни казалось многим это название, оно стало употребительным.

и элегантной теорией, имеющей неожиданные связи с некоторыми внешне далекими от оптимизации разделами современной математики.

• Современные пакеты для решения задач ЛП включают как методы внутренней точки, так и методы симплексного типа. Достаточно часто, в особенности на задачах большой размерности, методы внутренней точки демонстрируют лучшую производительность, чем симплекс-метод, и потому используются в этих пакетах по умолчанию. Таким образом, симплекс-метод хотя и продолжает активно использоваться, но стал все же скорее вспомогательным средством, тогда как основными методами решения задач ЛП стали практически и *теоретически* эффективные алгоритмы (счастье, возможно, и лучше правды, но правда и счастье вместе — это совсем хорошо). К этому следует добавить, что по многим оценкам в результате прогресса и в алгоритмах, и в компьютерах производительность ЛП-решателей выросла с начала 1980-х годов в 10^6 раз (!), причем три из этих шести порядков роста приходятся на прогресс в алгоритмах.

• Возникло замечательное обобщение ЛП — «коническое программирование». Первостепенную роль в здесь играют ЛП и два его «близких родственника» — программирование на конусах второго порядка (SOCP) и программирование на конусе неотрицательно определенных матриц (SDP). ЛП, SOCP и SDP имеют глубокое структурное сходство и допускают поэтому и общую теорию (прежде всего, чрезвычайно простую и удивительно полезную теорию двойственности), и сходные по устройству и весьма эффективные методы внутренней точки. В то же время «выразительные возможности» SOCP и SDP (т. е. круг оптимизационных задач, которые естественно сводятся к SOCP и SDP) существенно шире, чем (уже сами по себе достаточно широкие) «выразительные возможности» ЛП. В результате SOCP и в особенности SDP играют исключительно важную роль в построении и обработке (как численной, так и аналитической) оптимизационных моделей, возникающих в автоматическом регулировании, оптимизации структур, статистике, обработке сигналов, биоинформатике, решении задач дискретной оптимизации и во множестве других приложений.

В общем, плоды Interior Point Revolution, как говорится, трудно переоценить. Если бы меня спросили: «Кто построил эту дорогу», я упомянул бы многих коллег, но вряд ли назвал Леонида, а если бы и назвал, то далеко не в первую очередь. А вот если бы меня спросили, как получилось, что дорогу эту захотели и начали строить, я бы ответил, что все это сделал Леонид Хачиян, который «занимался величинами отвлеченно». А вышло приложение «к делу, к предметам»...

Биографическая справка

Леонид Генрихович Хачиян (3 мая 1952 — 29 апреля 2005) — российский математик. Предложил первый полиномиальный алгоритм — метод эллипсоидов — для решения задач линейного программирования. Этот результат произвел настоящую сенсацию и дал толчок к интенсивному поиску новых алгоритмов для решения задач линейного программирования.

Леонид Хачиян родился в Ленинграде. Его мама, Жанна Сааковна Саакян, проектировала многие сложные строительные объекты, например трансформаторную подстанцию в пос. Терскол, а отец, Генрих Борисович Хачиян (1913–2008), был офицером Военно-морского флота СССР, преподавал теоретическую механику в Московском институте химического машиностроения. В возрасте девяти лет Леонид вместе с родителями переехал в Москву. В 1969 г. поступил в Московский физико-технический институт (МФТИ). По окончании МФТИ в 1974 г. был принят на работу в Вычислительный центр АН СССР. В 1978 г. защитил кандидатскую диссертацию, а в 1984 г. — докторскую (обе — в Вычислительном центре АН СССР). В 1982 г. ему была присуждена Премия Фалкерсона за выдающиеся работы в области дискретной математики.С 1978 г. Леонид преподавал в МФТИ и читал лекции по теории алгоритмов. В конце 1989 г. Хачиян провел четыре месяца в качестве приглашенного профессора Корнельского университета, а с 1990 г. до своей безвременной кончины он был профессором университета Ратгерс (Rutgers).

В США Хачиян продолжал исследования по прежней тематике, а также вел исследования по новым направлениям. Последние 10 лет жизни Леонид вместе со своими коллегами и учениками интенсивно развивал новое направление: дуализация и задачи перечисления. Фактически это направление возникло из второго его наиболее известного результата — квазиполиномиального алгоритма для дуализации ДНФ, полученного вместе с М. Фредманом (M. Fredman) в 1995 г. Важный результат по этой тематике был получен Леонидом за месяц до смерти.

Раздел I

АЛГОРИТМЫ ДЛЯ МАТРИЧНЫХ ИГР

О скорости сходимости игровых процессов решения матричных игр[1)]

Аннотация. Показывается, что сходимость по числу итераций игровых процессов решения матричных (полиэдральных) игр независимо от выбора шагов на итерациях не может быть, вообще говоря, лучше гармонической. Приводятся оценки сходимости.

Для решения задач линейного программирования, особенно в тех случаях, когда размерность этих задач велика, в настоящее время достаточно часто используется метод, состоящий в сведении исходной задачи программирования к некоторой полиэдральной игре с последующим отысканием решения этой игры при помощи игровых итеративных процессов [1]. К достоинствам использования такого рода процессов, являющихся обобщением итеративного метода Брауна решения матричных игр, относятся бо́льшая свобода в формировании декомпозиционных алгоритмов, нечувствительность метода к случайным ошибкам вычислений, простота реализации процесса. В то же время вычислительный опыт показал быструю и надежную работу таких процессов лишь в том случае, когда требуемая точность невелика; по мере увеличения точности число итераций резко растет. Попытки ускорения сходимости путем использования различных вариантов выбора шагов на итерациях привели к мнению [1, 2], что для полиэдральных (матричных) игр сходимость игровых процессов не превышает гармонической вне зависимости от выбора шагов. Настоящая статья посвящена обоснованию этого факта и получению оценок сходимости, справедливых при любом выборе шагов. Эти оценки устанавливают предел, до которого возможно ускорение сходимости в классе рассмотренных в [1] игровых итеративных процессов.

В § 1 рассмотрен игровой процесс решения симметричной матричной игры и показано, что в том случае, когда все чистые стратегии игры существенны [3], имеет место справедливая вне зависимости от выбора шагов на итерациях оценка

$$u_k \geqslant u_0 \left(1 + \frac{u_0}{A} k\right)^{-1},$$

---

1) *Хачиян Л. Г.* О скорости сходимости игровых процессов решения матричных игр // ЖВМиМФ. 1977. Т. 17. № 6. С. 1421–1431.

где u_k — величина ошибки на k-й итерации, а A — некоторая (зависящая от игры) положительная константа. Отдельно рассмотрен практически важный случай игрового процесса в чистых стратегиях; в этом случае значение константы A выражено в явном виде через координаты седловой точки и коэффициенты матрицы платежа. Полученная оценка может достигаться при соответствующем выборе шагов на играх сколь угодно большой размерности.

Используя симметризацию игр по Нейману [4], результаты § 1 мы переносим в § 2 на случай произвольных (несимметричных) матричных игр. Поскольку любая полиэдральная игра представима в матричной форме, все результаты этого параграфа могут быть автоматически перенесены и на полиэдральные игры.

§ 1. Оценка снизу сходимости игрового процесса решения симметричной матричной игры

Рассмотрим симметричную матричную игру γ с кососимметрической $(n \times n)$-матрицей платежа $A = (a_{ij})$, $a_{ij} = -a_{ji}$, $i, j \in N = \{1, 2, \ldots, n\}$. Игрока, выбирающего строки матрицы, будем считать минимизирующим и обозначать через S стандартный симплекс его смешанных стратегий $x \in S$, а через e_i, $i \in N$ — его чистые стратегии. Условимся о следующем обозначении: если V — некоторый n-мерный вектор с компонентами V_i, $i \in N$, то через $\max(V)$ будет обозначена скалярная величина, равная покомпонентному максимуму, а через $|V|$ и $[V]^+$ обозначены n-мерные векторы с компонентами $|V_i|$ и $\max\{0, V_i\}$.

Положим $u(x) = \max(xA)$. Функция $u(x) \geqslant 0$ (индикатриса сходимости [1]) характеризует отклонение по цене точки $x \in S$ от множества S^* решений игры, т. е. от множества оптимальных стратегий $x^* \in S^*$, для которых $u(x^*) = 0$.

Игровой итеративный процесс [1] решения игры γ имеет вид

$$x_{k+1} = x_k(1 - \alpha_k) + \widehat{x}_k \alpha_k, \quad \widehat{x}_k \in \widehat{X}(x_k). \tag{1.1}$$

Здесь $k = 0, 1, \ldots$ — номер итерации, $x_k \in S$ — последовательные приближения, $\{\alpha_k\}$ — некоторая правильная [1] числовая последовательность

$$\alpha_k \in (0, 1], \quad \lim_{k\to\infty} \alpha_k = 0, \quad \sum_{k=0}^{\infty} \alpha_k = \infty, \tag{1.2}$$

а выпуклый компакт $\widehat{X}(x_k) \subseteq S$ задан при каждом $x_k \in S$ соотношением

$$\widehat{X}(x_k) = \operatorname{Arg} \max_{x \in S} x_k A x.$$

Известно [1], что в предположении (1.2) игровой процесс (1.1) сходится, т. е. $u(x_k) \to 0$ при $k \to \infty$ независимо от начального приближения $x_0 \in S$. Для оценки снизу быстроты сходимости к нулю величин $u(x_k)$ докажем предварительно следующую лемму.

Лемма 1. *Пусть $x^* \in S^*$ — какое-либо решение игры γ. Тогда для двух последовательных приближений x_k и x_{k+1} процесса* (1.1) *выполнена независимо от величины шага $\alpha_k \in (0, 1]$ оценка*

$$u\,(x_{k+1}) \geqslant u\,(x_k) \frac{|\widehat{x}_k A|\, x^*}{2u\,(x_k) + |\widehat{x}_k A|\, x^*}. \tag{1.3}$$

Доказательство. Умножая (1.1) справа на матрицу A, получим

$$x_{k+1} A = (1 - \alpha_k) x_k A + \alpha_k \widehat{x}_k A,$$

откуда следует

$$u(x_{k+1}) = \max(x_{k+1} A) = \max[(l - \alpha_k) x_k A + \alpha_k \widehat{x}_k A] \geqslant (1 - \alpha_k) x_k A y + \alpha_k \widehat{x}_k A y \tag{1.4}$$

для любого $y \in S$. Положим сначала $y = \widehat{x}_k$. Поскольку $\widehat{x}_k A \widehat{x}_k = 0$ (матрица A кососимметрическая) и

$$x_k A \widehat{x}_k = \max_{x \in S} x_k A x = \max(x_k A) = u(x_k),$$

то неравенство (1.4) в этом случае примет вид

$$u(x_{k+1}) \geqslant (1 - \alpha_k) u(x_k). \tag{1.5}$$

Для любого $i \in N$ имеем $u(x_k) \geqslant x_k A e_i$, и поэтому из (1.5) получаются n неравенств

$$u(x_{k+1}) \geqslant (1 - \alpha_k) x_k A e_i, \quad i \in N.$$

С другой стороны, из (1.4) при $y = e_i$ также следуют n неравенств

$$u(x_{k+1}) \geqslant (1 - \alpha_k) x_k A e_i + \alpha_k \widehat{x}_k A e_i, \quad i \in N.$$

Из совместного рассмотрения этих неравенств с предыдущими получим

$$u(x_{k+1}) \geqslant (1 - \alpha_k) x_k A e_i + \alpha_k \max\{0, \widehat{x}_k A e_i\}, \quad i \in N.$$

Умножим теперь i-е неравенство в получившейся системе на неотрицательную величину x_i^* (i-ю компоненту оптимальной стратегии $x^* \in S^*$) и просуммируем получившиеся неравенства по всем $i \in N$. В результате получится

$$u(x_{k+1}) \geqslant (1 - \alpha_k) x_k A x^* + \alpha_k [\widehat{x}_k A]^+ x^*.$$

Поскольку, в силу симметрии игры γ, оптимальная стратегия первого, минимизирующего, игрока является оптимальной и для второго, максимизирующего, игрока, то $x_k A x^* \geqslant 0$, и предыдущее неравенство можно переписать в виде

$$u(x_{k+1}) \geqslant \alpha_k [\widehat{x}_k A]^+ x^*.$$

Но

$$[\widehat{x}_k A]^+ x^* = \frac{1}{2}(|\widehat{x}_k A| x^* + \widehat{x}_k A x^*) \geqslant \frac{1}{2} |\widehat{x}_k A| x^*$$

(из оптимальности x^* вновь следует $\widehat{x}_k A x^* \geqslant 0$), и поэтому

$$u(x_{k+1}) \geqslant \frac{\alpha_k}{2} |\widehat{x}_k A| x^*.$$

Используя теперь (1.5), получаем

$$u(x_{k+1}) \geqslant \max\left\{ (1-\alpha_k) u(x_k),\ \frac{\alpha_k}{2} |\widehat{x}_k A| x^* \right\} \geqslant$$

$$\geqslant \min_{\alpha \in [0,1)} \max\left\{ (1-\alpha_k) u(x_k),\ \frac{\alpha_k}{2} |\widehat{x}_k A| x^* \right\} = u(x_k) \frac{|\widehat{x}_k A| x^*}{2u(x_k) + |\widehat{x}_k A| x^*},$$

что совпадает с (1.3). Лемма доказана. □

Пусть

$$\varphi(\widehat{x}) = \max_{x^* \in S^*} |\widehat{x} A| x^*. \tag{1.6}$$

Поскольку оценка (1.3) справедлива при любом $x^* \in S^*$ и максимум в (1.6) достигается, то в (1.3) можно величину $|\widehat{x}_k A| x^*$ заменить на $\varphi(\widehat{x}_k)$ и получить оценку

$$u(x_{k+1}) \geqslant u(x_k) \frac{\varphi(\widehat{x}_k)}{2u(x_k) + \varphi(\widehat{x}_k)}, \tag{1.7}$$

которой и будем пользоваться в дальнейшем. Следующая лемма устанавливает, что в том случае, когда величины $\varphi(\widehat{x}_k)$ ограничены снизу некоторым строго положительным числом, из (1.7) следует гармоническая оценка снизу на скорость убывания величин $u(x_k)$.

Лемма 2. *Пусть последовательность неотрицательных величин* $u(x_k) \geqslant 0$ *и последовательность ограниченных снизу некоторой положительной константой* c *величин* $\varphi(\widehat{x}_k)$

$$\varphi(\widehat{x}_k) \geqslant c > 0 \tag{1.8}$$

связаны при любом $k = 0,\ 1,\ \ldots$ *соотношением* (1.7). *Тогда справедлива оценка*

$$u(x_k) \geqslant u(x_0)\,[1 + 2ku(x_0)/c]^{-1}. \tag{1.9}$$

Доказательство. Из (1.7) и (1.8) следует

$$u(x_{k+1}) \geqslant u(x_k) \frac{c}{2u(x_k) + c}.$$

Положим $r_k = 2u(x_k)/c$ и перепишем предыдущее неравенство в виде

$$r_{k+1} \geqslant \frac{r_k}{1+r_k} = f(r_k), \quad k = 0, 1, \ldots, \tag{1.10}$$

где $f(r) = r/(1+r)$. Поскольку функция $f(r)$ монотонно возрастает по r, то из (1.10) следует

$$r_k \geqslant f(r_{k-1}) \geqslant f(f(r_{k-2})) \geqslant \ldots \geqslant \underbrace{f(f(\ldots(f(}_{k \text{ раз}}r_0))\ldots)).$$

Но

$$\underbrace{f(f(\ldots(f(}_{k \text{ раз}}r_0))\ldots)) = \frac{r}{1+kr},$$

и поэтому из предыдущей цепочки неравенств получим $r_k \geqslant r_0/(1+kr_0)$, что совпадает, в силу определения величин r_k, с оценкой (1.9). Лемма доказана. □

Из леммы 2 следует, что для того, чтобы оценить снизу сходимость величин $u(x_k)$ к нулю, достаточно оценить снизу последовательность величин $\varphi(\widehat{x}_k)$. Для игр, все стратегии которых существенны [3], такую оценку дает

Лемма 3. *Пусть у симметричной матричной игры γ все стратегии существенны. Тогда существует такая строго положительная константа $c(\gamma) > 0$, что для любой стратегии x, не являющейся решением игры (так что $u(x) > 0$), выполнено неравенство*

$$\min_{\widehat{x} \in \widehat{X}(x)} \varphi(\widehat{x}) \geqslant c(\gamma) > 0.$$

Доказательство. Напомним, что чистая стратегия e_i, $i \in N$ называется существенной, если существует такая оптимальная стратегия $x^* \in S^*$, что $x_i^* > 0$. Если у игры γ все чистые стратегии существенны, то в силу выпуклости множества S^* найдется [3] оптимальная стратегия $\bar{x}^* \in S^*$, у которой $\bar{x}_i^* > 0$ при всех $i \in N$.

Пусть 2^N — множество всех непустых подмножеств множества $N = \{1, 2, \ldots, n\}$ и $\omega \in 2^N$. Грань симплекса S, натянутую на чистые стратегии e_i, $i \in \omega$, обозначим через S_ω. Например, $S_N = S_{\{1,2,\ldots,n\}} = S$, $S_{(i)} = e_i$ и т. д., так что всего имеется $2^n - 1$ граней.

Для любого $x \in S$ компакт $\widehat{X}(x)$ представляет собой некоторую грань $S_{\omega(x)}$ симплекса S, причем множество $\omega(x)$ состоит из тех и только тех индексов $i \in N$, на которых достигается $\max(xA)$.

Положим

$$c_\omega = \min_{\widehat{x} \in S_\omega} \varphi(\widehat{x}), \quad \omega \in 2^N \tag{1.11}$$

и докажем, что если у игры γ все стратегии существенны и для некоторого $x \in S$ имеем $u(x) > 0$, то и $c_{\omega(x)} > 0$. В самом деле, если $c_{\omega(x)} = 0$, то, поскольку минимум в (1.11) достигается, найдется такой $\widehat{x}_0 \in \widehat{X}(x) = S_{\omega(x)}$, что

$$\varphi(\widehat{x}_0) = \max_{x^* \in S^*} |\widehat{x}_0 A| x^* = 0.$$

Тогда для вектора $\bar{x}^* \in S^*$, у которого все компоненты строго положительны, получим $|\widehat{x}_0 A| \bar{x}^* = 0$ и, значит, $\widehat{x}_0 A = 0$. Из равенства нулю вектора $\widehat{x}_0 A$ следует

$$0 = \widehat{x}_0 A x = -x A \widehat{x}_0 = -u(x),$$

что противоречит предположению $u(x) > 0$.

Теперь лемма доказывается следующим способом. Если среди всех смешанных стратегий $x \in S$ найдется такая, что $u(x) > 0$, то среди констант c_ω, $\omega \in 2^N$ найдется хотя бы одна строго положительная (например, $c_{\omega(x)}$); взяв в этом случае в качестве $c(\gamma) > 0$ минимальную из строго положительных констант c_ω, т. е. положив

$$c(\gamma) = \min_{\omega \in 2^N}{}^{+} c_\omega \tag{1.12}$$

(знак «$\min^+$» указывает, что минимум берется только по строго положительным величинам c_ω), получим для любого $x \in S$ такого, что $u(x) > 0$,

$$\min_{\widehat{x} \in \widehat{X}(x)} \varphi(\widehat{x}) = \min_{\widehat{x} \in S_{\omega(x)}} \varphi(\widehat{x}) = c_{\omega(x)} \geqslant c(\gamma) > 0,$$

что совпадает с утверждением леммы.

Если же для любой стратегии $x \in S$ имеем $u(x) = 0$, так что $S = S^*$, то матрица A нулевая; для $A = 0$ утверждение леммы выполнено (формально) при любой константе $c(0)$, в частности при $x(0) = 1$. Будем считать, что этот тривиальный случай также входит в (1.12), т. е. в определение знака «$\min^+$». Лемма доказана. □

Из лемм 2 и 3 вытекает, что сходимость игрового процесса (1.1) решения симметричной матричной игры γ, все стратегии которой существенны, не превышает гармонической.

Теорема 1. *Пусть γ — симметричная матричная игра, все стратегии которой существенны. Тогда независимо от выбора шагов $\alpha_k \in (0, 1]$ для игрового процесса* (1.1) *решения игры γ выполнена при любом $k = 0$, 1, … оценка* (1.9) *при $c = c(\gamma) > 0$, где константа $c(\gamma)$ задана соотношениями* (1.11) *и* (1.12).

Доказательство. При $u(x_0) = 0$ оценка (1.9) тривиально выполнена. Если же $u(x_0) > 0$, то из леммы 3 следует, что $\varphi(\widehat{x}_0) \geqslant c(\gamma) > 0$. Из (1.7) тогда получим, что и $u(x_1) > 0$. Рассуждая далее аналогичным образом, придем к выводу, что условие (1.8) выполнено при $c = c(\gamma) > 0$ для любого $k = 0, 1, \ldots$. Справедливость теоремы теперь следует из леммы 2. □

Замечание. При доказательстве теоремы второе и третье условия (1.2) использованы не были; теорема, таким образом, справедлива для любой последовательности шагов $\alpha_k \in (0, 1]$, в частности для нерегулярных [1] игровых процессов.

Рассмотрим теперь важный с практической точки зрения частный случай, когда в игровом процессе (1.1) решения симметричной матричной игры γ в качестве векторов $\widehat{x}_k \in \widehat{X}(x_k)$ выбираются лишь чистые стратегии $e_{i_k} \in \widehat{X}(x_k)$, т. е. когда игровой процесс имеет вид

$$x_{k+1} = x_k(1-\alpha_k) + e_{i_k}\alpha_k, \quad e_{i_k} \in \widehat{X}(x_k). \tag{1.1$'$}$$

Процесс (1.1) будем называть игровым процессом в чистых стратегиях. В случае игрового процесса в чистых стратегиях оценка (1.7) может быть записана в виде

$$u(x_{k+1}) \geqslant u(x_k)\frac{\varphi(e_{i_k})}{2u(x_k)+\varphi(e_{i_k})}, \tag{1.7$'$}$$

где

$$\varphi(e_i) = \max_{x^* \in S^*} \sum_{j \in N} |a_{ij}| x_j^*, \quad i \in N.$$

Отметим, что величина $\varphi(e_i)$ может быть интерпретирована как средняя абсолютная величина платежа, который при разыгрывании игры выплачивается каким-либо партнером другому в том случае, когда один из партнеров придерживается чистой стратегии e_i, а другой — какой-либо оптимальной стратегии $x^* \in S^*$.

Если у игры γ все стратегии существенны, то из условия $u(x_k) > 0$ следует, в силу леммы 3, что $c_{\omega(x_k)} > 0$ и, значит, тем более $\varphi(e_{i_k}) \geqslant c_{\omega(x_k)} > 0$. Поэтому из (1.7′) и леммы 2 получается следующая теорема о сходимости игрового процесса в чистых стратегиях (1.1′).

Теорема 2. *Пусть γ — симметричная матричная игра, все стратегии которой существенны. Тогда независимо от выбора шагов $\alpha_k \in (0, 1]$ для игрового процесса в чистых стратегиях* (1.1′) *выполнена при любом $k = 0, 1, \ldots$ оценка* (1.9) *с константой*

$$c = c_e(\gamma) = \min_{i \in N}{}^{+} \varphi(e_i) > 0.$$

Доказательство теоремы 2 совершенно аналогично доказательству теоремы 1. □

Замечание. Пусть симметричная матричная игра γ^n, $n = 3, 4, \ldots$ задана $(n \times n)$-матрицей платежа

$$A^n = \begin{pmatrix} \begin{matrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{matrix} & 0 \\ 0 & 0 \end{pmatrix}$$

Множество S^* решений игры γ^n состоит из тех и только тех смешанных стратегий $x^*=(x_1^*, \dots, x_n^*)$, у которых совпадают первые три компоненты; поэтому все чистые стратегии игры γ^n существенны и $c_e(\gamma^n)=2/3$. Если в качестве начального приближения x_0 для процесса (1.1′) взять вектор $x_0=(2/3, 0, 1/3, 0, \dots, 0)$, так что $u(x_0)=1/3$, а величину шага α_k определять из условия

$$\alpha_k=\frac{3u(x_k)}{1+3u(x_k)}, \quad k=0, 1, \dots ,$$

то, воспользовавшись равенством

$$xA^ne_1+xA^ne_2+xA^ne_3=0$$

и тем фактом, что чистые стратегии $e_1, \dots, e_n$ не участвуют в процессе (1.1′), можно показать, что

$$u(x_k)=u(x_0)\left(1+\frac{2u(x_0)}{c_e(\gamma^n)}k\right)^{-1}=\frac{1}{3(k+1)}.$$

Таким образом, оценка (1.9) с константой $c=c_e(\gamma)$ может нетривиальным образом ($u(x_0)>0$, $n\neq 1, 2$) достигаться для процесса (1.1′) на играх любой размерности.

Теоремы 1 и 2 касались вопросов сходимости игровых процессов (1.1) и (1.1′) решения симметричных матричных игр, все стратегии которых существенны. Разумеется, если для произвольной симметричной матричной игры γ окажется, что

$$\varphi(\gamma)=\min_{i\in N}\varphi(e_i)>0 \tag{1.13}$$

(минимум берется по всем, а не только по строго положительным величинам $\varphi(e_i)$), то в силу (1.7′) и леммы 2 для процесса (1.1′) решения игры γ будет выполнена оценка (1.9) с константой $c=\varphi(\gamma)$. Таким образом, справедлива

Теорема 3. *Пусть для произвольной симметричной матричной игры γ выполнено условие* (1.13). *Тогда независимо от выбора шагов $\alpha_k\in(0, 1]$ для игрового процесса в чистых стратегиях* (1.1′) *решения игры γ выполнена при любом $k=0, 1, \dots$ оценка* (1.9) *с константой $c=\varphi(\gamma)$, задаваемой* (1.13).

З а м е ч а н и е. Можно показать, что необходимым условием для строгой положительности $\varphi(\gamma)$ является отсутствие у игры γ решений в чистых стратегиях. Это условие, однако, не является достаточным.

В заключение этого раздела отметим, что вопрос о том, может ли игровой процесс (1.1′) иметь скорость сходимости, превышающую гармоническую, для игр, у которых отсутствует решение в чистых стратегиях, выяснить не удалось. По-видимому, ответ на этот вопрос должен быть отрицательным. Во всяком случае, обращение в нуль хотя бы одной из величин $\varphi(e_i)$, $i\in N$, означает, в предположении отсутствия у игры решений в чистых стратегиях, обращение в нуль хотя бы одного из $n(n-1)/2$ независимых коэффициентов кососимметрической матрицы платежа игры. Поэтому для игр, у которых отсутствует решение в чистых стратегиях (рассматриваемых как точки $[n(n-1)/2]$-мерного пространства, задаваемые матрицами платежа), сходимость процесса (1.1′) не превышает гармонической почти всюду.

§ 2. Случай произвольной (несимметричной) матричной игры

В этом параграфе результаты, полученные ранее для симметричных матричных игр, будут перенесены па случай произвольных (несимметричных) матричных игр. Для этого, используя симметризацию игр по Нейману [4], общий случай сведем к симметричному, а затем используем теоремы 1—3.

Пусть Г — произвольная матричная игра, задаваемая матрицей платежа $B=(b_{ij})$, $i\in N=\{1, 2, \dots, n\}$, $j\in M=\{1, 2, \dots, m\}$. Игрока, выбирающего строки матрицы B, будем по-прежнему считать минимизирующим и обозначать через S_n стандартный симплекс его смешанных стратегий $y\in S_n$ размерности n, а через e_i, $i\in N$ — чистые стратегии этого игрока. Аналогично, второй игрок — максимизирующий, S_m — стандартный симплекс его смешанных стратегий $z\in S_m$ и e_j, $j\in M$ — его чистые стратегии.

Положим $V(y, z)=\max(yB)-\min(Bz)$. Функция $V(y, z)\geqslant 0$ (индикатриса сходимости [1]) характеризует отклонение по цене точки (y, z) от множества $S_n^*\times S_m^*$ решений игры Г, т. е. от множества седловых точек $(y^*, z^*)\in S_n^*\times S_m^*$ таких, что $V(y^*, z^*)=0$.

Игровой итеративный процесс решения игры Г имеет вид [1]

$$\begin{aligned} y_{k+1} &= y_k(1-\alpha_k)+\widehat{y}_k\alpha_k,\ \widehat{y}_k\in\widehat{Y}(z_k)=\operatorname{Arg}\min_{y\in S_n} yBz_k,\\ z_{k+1} &= z_k(1-\alpha_k)+\widehat{z}_k\alpha_k,\ \ \widehat{z}_k\in\widehat{Z}(z_k)=\operatorname{Arg}\max_{z\in S_m} y_kBz. \end{aligned} \tag{2.1}$$

Здесь, как и в (1.1), $k=0, 1, \dots$ — номер итерации, (y_k, z_k) — последовательные приближения и $\{\alpha_k\}$ — некоторая правильная числовая последовательность (1.2).

Для того чтобы свести игровой процесс (2.1) к игровому процессу решения некоторой симметричной игры, введем следующие обозначения. Для n-мерного вектора $y=(y_1, \dots, y_i, \dots, y_n)$ и m-мерного вектора $z=(z_1, \dots, z_j, \dots, z_m)$ через $x=y*z$ обозначим nm-мерный вектор x с координатами $x_{(ij)}=y_iz_j$, $(i, j)\in N\times M$. Очевидно, что если $y\in S_n$ — некоторая смешанная стратегия размерности n и $z\in S_m$ — некоторая смешанная стратегия размерности m, то nm-мерный вектор $x=y*z$ также является смешанной стратегией $x\in S_{nm}$. Отметим также, что отображение $y*z\to x$ взаимно-однозначно отображает $S_n\times S_m$ в S_{nm}.

Пусть, далее, $A=(a_{(\alpha\beta)(ij)})$ — кососимметрическая $(nm \times nm)$-матрица платежа симметричной игры γ_Γ, полученной в результате симметризации игры Г, по Нейману:

$$a_{(\alpha\beta)(ij)}=b_{\alpha j}-b_{i\beta}, \quad (\alpha, \beta), (i, j)\in N\times M. \tag{2.2}$$

Положим, как и в § 1, для любого $x\in S_{nm}$

$$u(x)=\max(xA), \quad \widehat{X}(x)=\operatorname{Arg}\max_{\widehat{x}\in S_{nm}} xA\widehat{x}.$$

Поскольку для любой чистой стратегии $e_{(i,j)}\in S_{nm}$ и любых смешанных стратегий $y\in S_n$ и $z\in S_m$ из (2.2) следует

$$(y*z)Ae_{(i,j)}=yBe_j-e_iBz, \quad (i, j)\in N\times M,$$

то справедливы соотношения

$$u(y*z)=V(y, z), \tag{2.3}$$

$$\widehat{y}*z\in\widehat{X}(y*z) \quad \text{для любых } \widehat{y}\in\widehat{Y}(z) \text{ и } \widehat{z}\in\widehat{Z}(y). \tag{2.4}$$

Вернемся теперь к игровому процессу (2.1). Если положить $x_k= =y_k*z_k$, то из (2.1) следует

$$x_{k+1}=y_{k+1}*z_{k+1}=[(1-\alpha_k)y_k+\alpha_k\widehat{y}_k]*[(1-\alpha_k)z_k+\alpha_k\widehat{z}_k]=$$
$$=(1-\alpha_k)y_k*z_k+\alpha_k\widehat{y}_k*z_k-\alpha_k(1-\alpha_k)(\widehat{y}_k-y_k)*(\widehat{z}_k-z_k).$$

Используя (2.4), перепишем предыдущее равенство в виде

$$x_{k+1}=x_k(1-\alpha_k)+\widehat{x}_k\alpha_k+\xi_k, \quad \widehat{x}_k\in\widehat{X}(x_k), \tag{2.5}$$

где через ξ_k обозначен nm-мерный вектор

$$\xi_k=-\alpha_k(1-\alpha_k)(\widehat{y}_k-y_k)*(\widehat{z}_k-z_k).$$

Заметим теперь, что при всех $k=0,\ 1,\ \ldots$ имеем $\xi_kA=0$. Поэтому, поскольку как компакт $\widehat{X}(x_k)$, так и величина $u(x_k)$ зависят на каждой итерации лишь от вектора x_kA (а не непосредственно от смешанной стратегии x_k), то значения величины $u(x_k)$, которые могут реализовываться на k-й итерации процесса (2.5) при заданном начальном приближении $x_0=y_0*z_0$, в точности совпадают со значениями величины $u(x_k)$, которые могут реализовываться на k-й итерации игрового процесса

$$x_{k+1}=x_k(1-\alpha_k)+\widehat{x}_k\alpha_k, \quad \widehat{x}_k\in\widehat{X}(x_k), \tag{2.6}$$

решения симметризованной, по Нейману, игры γ_Γ при том же начальном приближении x_0. Для того чтобы теперь, в соответствии с равенством $V(y_k, z_k)=u(y_k*z_k)=u(x_k)$, применить к процессу (2.6) теоремы 1—3, осталось лишь остановиться на следующих двух обстоятельствах.

Если $S_n^* \times S_m^*$ — множество решений исходной игры Γ, а S_{nm}^* — множество решений симметризованной, по Нейману, игры γ_Γ, то из (2.3) следует, что $S_n^* * S_m^* \subseteq S_{nm}^*$. Отсюда, в частности, следует, что если у игры Γ все стратегии существенны, то и у игры γ_Γ все стратегии существенны.

Если игровой процесс (2.1) решения игры Γ является игровым процессом в чистых стратегиях

$$\begin{aligned} y_{k+1} &= y_k(1-\alpha_k) + e_{i_k}\alpha_k,\ e_{i_k} \in \widehat{Y}(z_k), \\ z_{k+1} &= z_k(1-\alpha_k) + e_{j_k}\alpha_k,\ e_{j_k} \in \widehat{Z}(z_k), \end{aligned} \tag{2.1$'$}$$

то, поскольку $\widehat{x}_k = \widehat{y}_k * \widehat{z}_k = e_{i_k} * e_{j_k} = e_{(i_k j_k)}$ в (2.6) — вновь чистая стратегия, получающийся в результате сведения (2.1′) к игровому процессу решения игры γ_Γ процесс (2.6) также является игровым процессом в чистых стратегиях.

Теперь можно сформулировать теоремы о сходимости игровых процессов решения произвольных (несимметричных) матричных игр.

Теорема 4. *Пусть Γ — матричная игра, все стратегии которой существенны. Тогда независимо от выбора шагов $\alpha_k \in (0, 1]$ для игрового процесса* (2.1) *решения игры Γ выполнена при любом $k = 0, 1, \ldots$ оценка*

$$V(y_k, z_k) \geqslant V(y_0, z_0)[1 + 2kV(y_0, z_0)/c]^{-1} \tag{2.7}$$

с некоторой строго положительной константой $c = c(\Gamma) > 0$.

Теорема 5. *Определим для матричной игры Γ с $(n \times m)$-матрицей платежа $B = (b_{ij})$ величины $\varphi(e_i, e_j)$ равенствами*

$$\varphi(e_i, e_j) = \max_{(y^*, z^*) \in S_n^* \times S_m^*} \sum_{(\alpha, \beta) \in N \times M} y_\alpha^* z_\beta^* |b_{i\beta} - b_{\alpha j}|.$$

Тогда если у игры Γ все стратегии существенны, то независимо от выбора шагов $\alpha_k \in (0, 1]$ для игрового процесса в чистых стратегиях (2.1′) *решения игры при любом $k = 0, 1, \ldots$ справедлива оценка* (2.7) *с константой*

$$c = c_e(\Gamma) = \min_{(i,j) \in N \times M}{}^{+} \varphi(e_i, e_j) > 0.$$

Теорема 6. *Пусть для произвольной матрицы игры Γ с $(n \times m)$-матрицей платежа $B = (b_{ij})$ выполнено условие*

$$\varphi(\Gamma) = \varphi(\gamma_\Gamma) = \min_{(i,j) \in N \times M} \varphi(e_i, e_j) > 0.$$

Тогда независимо от выбора шагов $\alpha_k \in (0, 1]$ для игрового процесса в чи-

стых стратегиях (2.1′) *решения игры* Γ *выполнена при любом* $k = 0, 1, \ldots$ *оценка* (2.7) *с константой* $c = \varphi(\Gamma)$.

Так же, как и в § 1, из теоремы 6 следует, что для игр Γ, у которых отсутствует решение в чистых стратегиях (рассматриваемых как точки nm-мерного пространства, задаваемые матрицами платежа), сходимость процесса (2.1′) не превышает гармонической почти всюду.

В заключение отметим следующее обстоятельство. Все сказанное выше относилось к игровым процессам с одинаковым выбором шагов вдоль двойственных направлений. Если рассматривать игровые процессы с неодинаковым выбором шагов вдоль двойственных направлений, т. е., например, процессы вида

$$\begin{aligned} y_{k+1} &= y_k(1-\alpha_k) + e_{i_k}\alpha_k, \quad e_{i_k} \in \widehat{Y}(z_k), \\ z_{k+1} &= z_k(1-h_k) + e_{j_k}h_k, \quad e_{j_k} \in \widehat{Z}(z_k) \end{aligned} \tag{2.8′}$$

при $\alpha_k \neq h_k$, то на процессы такого рода выводы предыдущих параграфов не распространяются. Так, например, у матричной игры Γ^0 с (2×2)-матрицей платежа

$$B = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

все стратегии существенны, и поэтому сходимость процесса (2.1) (и тем более процесса (2.1′)) с одинаковым выбором шагов вдоль двойственных направлений не превышает гармонической. В то же время для игры Γ^0 легко строится алгоритм вида (2.8′) с ограниченным отношением шагов α_k и h_k вдоль двойственных направлений

$$q \leqslant \frac{\alpha_k}{h_k} \leqslant \frac{1}{q}, \qquad 0 < q < 1,$$

сходимость которого по числу итераций экспоненциальна: $V(y_k, z_k) \leqslant \leqslant V(y_0, z_0)\, q^{k-1}$.

Автор благодарен А. И. Эрлиху за советы и ценные замечания.

Литература

[1] Итеративные методы в теории игр и программировании / под ред. В. З. Беленького, В. А. Волконского. М. : Наука, 1974.

[2] *Гольштейн Е. Г., Юдин Д. Б.* Новые направления в линейном программировании. М. : Сов. радио, 1966.

[3] *Гейл Д., Шерман С.* Решение конечных игр двух лиц // Матричные игры. М. : Физматгиз, 1961. С. 45—61.

[4] *Гейл Д., Кун Х. У., Таккер А. У.* О симметричных играх // Матричные игры. М. : Физматгиз, 1961. С. 62—71.

Сублинейный приближенный вероятностный алгоритм для матричных игр[1]

Аннотация. В статье строится параллельный рандомизированный алгоритм для вычисления пары ε-оптимальных стратегий для заданной (m, n)-матричной игры с матрицей платежа $A=(a_{ij})\in[-1, 1]^{m\times n}$ со средним временем работы $O(\varepsilon^{-2}\log^2(n+m))$ на EREW PRAM с $(n+m)/\log(n+m)$ процессорами. При любой фиксированной точности $\varepsilon>0$ среднее время работы процедуры равно $O((n+m)\log(n+m))$, т. е. сублинейно по длине входа mn. С другой стороны, приводятся простые аргументы, показывающие, что для $\varepsilon<1/2$ любой детерминированный алгоритм вычисления пары ε-оптимальных стратегий в (m, n)-матричной игре A с матрицей из ± 1 обязан просматривать $\Omega(mn)$ элементов A. Таким образом, при $m=n$ вероятностный алгоритм дает почти квадратичное ускорение по сравнению с детерминированным.

Ключевые слова: *приближенный алгоритм*, *сложность*; *матричные игры*, *параллельные алгоритмы*; *вероятностные алгоритмы*.

§ 1. Введение

Пусть $A=[a_{ij}]$ — (m, n)-*матричная игра* и

$$
\begin{aligned}
v^* &= \min\{v \,|\, Ax \leqslant ve,\ e^T x = 1,\quad x \geqslant 0\} = \\
&= \max\{v \,|\, A^T y \geqslant ve,\ e^T y = 1,\quad y \geqslant 0\} \text{ —}
\end{aligned}
\tag{1.1}
$$

ее цена, которая всегда достигается на некоторой паре *оптимальных стратегий* $x^*\in\mathbb{R}^n$ и $y^*\in\mathbb{R}^m$. Без ограничения общности считаем, что элементы матрицы платежа A нормализованы и лежат в фиксированном интервале, скажем, $[-1, 1]$.

Назовем пару стратегий *ε-оптимальной* для A, если

$$
\begin{aligned}
Ax &\leqslant (v^*+\varepsilon)e,\quad e^T x = 1,\quad x \geqslant 0,\\
A^T y &\geqslant (v^*-\varepsilon)e,\quad e^T y = 1,\quad y \geqslant 0,
\end{aligned}
\tag{1.2}
$$

где ε — заданная (аддитивная) абсолютная точность вычисления, а e — это вектор, все компоненты которого единицы. В [3] был построен детерминированный параллельный приближенный алгоритм, вычисляющий ε-оптимальное решение за время $O(\varepsilon^{-2}\log^2(n+m))$ на mn EREW PRAM процессорах. С одной стороны, алгоритм использует обоб-

[1]) *Grigoriadis M.*, *Khachiyan L.* A sublinear-time randomized approximation algorithm for matrix games // Oper. Res. Lett. 1995. V. 18. No. 2. P. 53—58. Перевод С. П. Тарасова.

щение описанного в [4] метода экспоненциального потенциала для задач выпуклой блочной оптимизации, с другой, — он может быть получен с некоторым ухудшением оценок из работы Плоткина и др. [8]. Недавно Луби и Нисан [7] предложили параллельный приближенный алгоритм для решения задачи $\min\{c^T x \mid Ax \geqslant b,\ x \geqslant 0\}$ с неотрицательными входными данными. Эту задачу принято называть «положительное линейное программирование», и она эквивалентна матричным играм. Алгоритм из [7] решает (m, n)-матричную игру с матрицей $A \geqslant 0$ с *относительной* точностью ε за параллельное время $O(\varepsilon^{-4} \log n \log(n+m) \log(m/\varepsilon))$ на mn EREW PRAM процессорах[1]).

В настоящей статье мы предлагаем приближенный вероятностный алгоритм для матричный игр, вычисляющий пару ε-оптимальных стратегий для данной игры $A \in [-1, 1]^{mn}$ со средним параллельным временем работы $O(\varepsilon^{-2} \log^2(n+m))$ и использующий только $(n+m)/\log(n+m)$ EREW PRAM процессоров. Для любого фиксированного $\varepsilon > 0$ среднее время работы последовательного варианта нашей процедуры равно $O((n+m) \log(n+m))$ и является сублинейным по числу элементов входа A, если, скажем, величины m и n имеют одинаковый порядок. Мы также показываем, что для $\varepsilon < 1/2$ любая детерминированная приближенная процедура вычисления пары ε-оптимальных стратегий (m, n)-матричной игры с матрицей из ± 1 обязана просматривать почти половину элементов A.

Быстрые параллельные алгоритмы для матричных игр и положительного линейного программирования могут использоваться в качестве эвристических процедур в быстрых приближенных алгоритмах генерации столбцов, например, при решении задачи раскроя или для алгоритмов типа Данцига—Вульфа решения блочных задач линейного программирования и проч. Двойственные задачи (1.1) образуют также двойственную пару задач дробной упаковки и дробного покрытия вершин в графе (см. [8]). Дополнительно, наш алгоритм строит пару ε-оптимальных стратегий, являющихся выпуклой комбинацией $O(\varepsilon^{-2} \log^2(n+m))$ чистых стратегий, и для любого фиксированного $\varepsilon > 0$ это число растет лишь логарифмически от общего количества чистых стратегий. Такие ε-оптимальные стратегии имеют различные приложения в теории сложности [6]. Их существование также следует из описания детерминированного параллельного приближенного алгоритма, предложенного в [3].

Наш вероятностный алгоритм навеян классическим итеративным алгоритмом Брауна решения матричных игр [1], который известен как

[1]) См. также работу: *Фомин С. В.* Быстрый приближенный алгоритм для задачи положительного линейного программирования // Математические методы и алгоритмы. Т. 6. М. : ИСП РАН. 2005. С. 27—40. — *Примеч. пер.*

метод *фиктивного разыгрывания*:

$$
\begin{aligned}
&X(0)=0\in\mathbb{R}^n,\quad Y(0)=0\in\mathbb{R}^m.\quad \text{Для } t=0,\,1,\,\ldots,\ \text{повторять}\\
&\quad X(t+1)=X(t)+e_i(t),\ \ e_i(t)=\operatorname{argmin}\{Y(t)Ae_i\ \ |\, i=1,\,\ldots,\,n\},\\
&\quad Y(t+1)=Y(t)+e_j(t),\ \ e_j(t)=\operatorname{argmax}\{e_j^T AX(t)\,|\, j=1,\,\ldots,\,m\},
\end{aligned}
\tag{1.3}
$$

где e_i и e_j обозначают i-й и j-й орты в $\mathbb{R}^n$ и $\mathbb{R}^m$, соответственно. Сходимость процесса (1.3) для произвольной матричной игры A

$$
\begin{gathered}
A\frac{X(t)}{t}\leqslant(v^*+\varepsilon(t))e,\quad e^T\frac{X(t)}{t}\ =1,\quad \frac{X(t)}{t}\geqslant 0,\\
A^T\frac{Y(t)}{t}\geqslant(v^*-\varepsilon(t))e,\quad e^T\frac{Y(t)}{t}e=1,\quad \frac{Y(t)}{t}\geqslant 0,\\
\varepsilon(t)\to 0 \text{ при } t\to\infty
\end{gathered}
$$

была показана Робинсон [9]. Единственная известная верхняя оценка скорости сходимости метода $\varepsilon(t)\leqslant 2^{n+m}t^{-1/(n+m-2)}$ получена Шапиро [10]. Нижняя оценка скорости сходимости более широкого класса методов получена Хачияном [5]. Согласно старой гипотезе оценка Шапиро слишком завышена, а истинное число фиктивных разыгрываний (итераций) должно расти квадратично по $1/\varepsilon$.

На t-й итерации метода Брауна — Робинсон минимизирующий игрок вычисляет индекс $i(t)$ минимальной компоненты вектора $V(t)= Y(t)\,A\in\mathbb{R}^n$ (см. вторую строчку (1.3)) и затем увеличивает на единицу $i(t)$-ю компоненту $X(t)$. Иначе говоря, минимизирующий игрок делает ход в направлении чистой стратегии $e_{i(t)}\in\mathbb{R}^n$, которая минимизирует его платеж для текущей стратегии максимизирующего игрока $Y(t)$. В это же самое время максимизирующий игрок выбирает свою чистую стратегию $e_{j(t)}\in\mathbb{R}^m$, которая максимизирует его текущий платеж $e_j^T U(t)$, где $U(t)=AX(t)\in\mathbb{R}^m$. Для симметричных игр ($A=-A^T$) итеративная процедура (1.3) порождает $X(t)=Y(t)$ и $V(t)=-U(t)$ для всех t.

Наш алгоритм является простой вероятностной модификацией детерминированного метода, в котором индексы $i(t)\in\{1,\,\ldots,\,n\}$ и $i(t)\in\{1,\,\ldots,\,n\}$ выбираются с вероятностью, пропорциональной $\exp\{-\varepsilon\times V_i(t)/2\}$ или $\exp\{+\varepsilon U_j(t)/2\}$ для минимизирующего или максимизирующего игрока, соответственно. Такое распределение сосредоточено на чистых стратегиях, которые были бы выбраны в процедуре Брауна — Робинсон, но свободно от ее жесткого разрывного характера.

В §§ 2 и 3 будет проанализирована трудоемкость такого вероятностного алгоритма. А в последнем параграфе мы приведем простую нижнюю теоретико-информационную оценку сложности детерминированных приближенных алгоритмов для матричных игр.

§ 2. Параллельный вероятностный алгоритм для симметричных игр

Рассмотрим сначала *симметричную* игру, для которой $A=-A^T$. В этом случае $m=n$, цена игры $v^*=0$, множества оптимальных стратегий для обоих игроков совпадают, а (1.2) сводится к вычислению такого x, что

$$Ax \leqslant \varepsilon e, \qquad x \in S=\{x \in \mathbb{R}^n \mid e^T x=1,\ x \geqslant 0\}. \tag{2.1}$$

Следующий алгоритм с вероятностью не менее $1/2$ находит ε-оптимальную стратегию x для A за параллельное время $O(\varepsilon^{-2}\log^2(n))$ на EREW PRAM с $n/\log n$ процессорами. Ниже без ограничения общности считаем, что $\varepsilon \in (0, 1]$ и $n \geqslant 8$.

АЛГОРИТМ $\mathscr{A}$

1 **Инициализация:** $X \leftarrow U \leftarrow 0 \in \mathbb{R}^n$, $p \leftarrow \dfrac{e}{n}$, $t \leftarrow 0$.

2 **Повторить:**

3 **Счетчик итераций:** $t \leftarrow t+1$.

4 **Датчик случайных чисел:** выбираем $k \in \{1, \ldots, n\}$ с вероятностью p_k.

5 **Модификация X:** $X_k \leftarrow X_k+1$.

6 **Модификация U:** для $i=1, \ldots, n$ параллельно выполнить $U_i \leftarrow U_i+a_{ik}$.

7 **Модификация p:** для $i=1, \ldots, n$ параллельно выполнить

$$p_i \leftarrow \frac{p_i \exp\left\{\dfrac{\varepsilon}{2}a_{ik}\right\}}{\sum\limits_{j=1}^{n} p_j \exp\left\{\dfrac{\varepsilon}{2}a_{jk}\right\}}.$$

8 **Критерий останова:** если $U/t \leqslant \varepsilon e$, то останавливаемся и печатаем $x=X/t$. □

Теорема 1.

i) *Алгоритм $\mathscr{A}$ корректный: критерий останова в строке 8 описания гарантирует ε-оптимальность выходного вектора x.*

ii) *С вероятностью $\geqslant 1/2$ алгоритм $\mathscr{A}$ остановится через $t^*=4\varepsilon^{-2}$ итераций.*

Доказательство. В строках 5 и 6 описания поддерживаются инварианты $\dfrac{X(t)}{t} \in S=\{x \in \mathbb{R}^n \mid e^T x=1,\ x \geqslant 0\}$ и $U(t)=AX(t)$, где $X(t)$, $U(t)$ обозначают n-векторы X, U в конце итерации $t \geqslant 1$. Отсюда сразу следует утверждение (i) теоремы.

Из строк 6 и 7 описания алгоритма $\mathscr{A}$ легко вытекает, что

$$p_i(t)=\frac{P_i(t)}{\sum\limits_{j=1}^{n} P_j(t)}, \qquad \text{для } t=1, 2, \ldots,$$

где

$$P_i(t)=\exp\{\frac{\varepsilon U_i(t)}{2}\}, \qquad i=1, \ldots, n. \tag{2.2}$$

Введем потенциальную функцию

$$\Phi(t) = \sum_{i=1}^{n} P_i(t). \tag{2.3}$$

Если на итерации t в строке 4 описания выбирается $k \in \{1, \ldots, n\}$, то

$$\Phi(t+1) = \sum_{i=1}^{n} P_i(t+1) = \sum_{i=1}^{n} \exp\{\frac{\varepsilon U_i(t+1)}{2}\} = \sum_{i=1}^{n} \exp\{\frac{\varepsilon U_i(t)}{2} + \frac{\varepsilon a_{ik}}{2}\} =$$

$$= \sum_{i=1}^{n} P_i(t) \exp\{\frac{\varepsilon a_{ik}}{2}\} = \Phi(t) \sum_{i=1}^{n} p_i(t) \exp\{\frac{\varepsilon a_{ik}}{2}\},$$

и поскольку k выбирается с вероятностью $p_k(t)$, то мы заключаем, что математическое ожидание $\Phi(t+1)$ дается формулой

$$\mathsf{E}[\Phi(t+1)] = \Phi(t) \sum_{i,k=1}^{n} p_i(t)\, p_k(t) \exp\{\frac{\varepsilon a_{ik}}{2}\}. \tag{2.4}$$

Из условия $a_{ik} \in [-1, 1]$ получаем, что для всех $\varepsilon \in (0, 1]$

$$\exp\{\frac{\varepsilon a_{ik}}{2}\} \leqslant 1 + \frac{\varepsilon}{2} a_{jk} + \frac{\varepsilon^2}{6}.$$

Из кососимметричности A получаем

$$\sum_{i,\,k=1}^{n} p_i(t)\, p_k(t)\, a_{ik} = 0.$$

Кроме того, для любого $p(t) \in S$ выполнено

$$\sum_{i,\,k=1}^{n} p_i(t) p_k(t) = \left(\sum_{i=1}^{n} p_i(t)\right)^2 = 1.$$

Следовательно,

$$\mathsf{E}[\Phi(t+1)] \leqslant \mathsf{E}[\Phi(t)] \left(1 + \frac{\varepsilon^2}{6}\right),$$

что сразу влечет

$$\mathsf{E}[\Phi(t)] \leqslant \Phi(0) \left(1 + \frac{\varepsilon^2}{6}\right)^t = n\left(1 + \frac{\varepsilon^2}{6}\right)^t \leqslant n \exp\left\{\frac{t\varepsilon^2}{6}\right\}.$$

В частности, после $t^* = 4\varepsilon^{-2} \ln n$ итераций получаем $\mathsf{E}[\Phi(t^*)] \leqslant n^{5/3}$, что по неравенству Маркова влечет выполнение неравенства $\Phi(t^*) \leqslant 2n^{5/3}$ с вероятностью не меньше $1/2$. И поскольку мы считаем $n \geqslant 8$, то

$$\Phi(t^*) \leqslant n^2 \quad \text{с вероятностью} \ \geqslant \frac{1}{2}.$$

С другой стороны, если $\Phi(t^*) \leqslant n^2$, то из определения (2.3) потенциа-