ЭКОНОМИКА И МАТЕМАТИЧЕСКИЕ МЕТОДЫ, 2012, том 48, № 4, с. 113–118

## МЕТОДЫ ОПТИМИЗАЦИИ

# ДВОЙСТВЕННЫЙ МЕТОД РЕШЕНИЯ БИМАТРИЧНЫХ ИГР

© 2012 г. А.А. Заславский, А.Н. Липатова

*(Москва)*

Работа посвящена исследованию двойственного метода нахождения равновесия в биматричной игре. Хотя теоретического доказательства сходимости метода получить не удалось, проведенные вычислительные эксперименты показали, что он может использоваться для решения биматричных игр, возможно, в сочетании с другими методами.

**Ключевые слова:** биматричные игры, линейное программирование, двойственная задача.

## 1. ПОСТАНОВКА ЗАДАЧИ

Биматричной игрой называется конечная бескоалиционная игра двух лиц. Ее можно описать системой $\Gamma = \langle I, J, h_1, h_2\rangle$, где $I = \{1,..., m\}$, $J = \{1,..., n\}$ – множества чистых стратегий игроков 1 и 2 соответственно. Если ввести для значений функций выигрыша на каждой паре стратегий обозначения $h_1(i, j) = a_{ij}$, $h_2(i, j) = b_{ij}$, то вся информация об игре будет задаваться матрицами $A = (a_{ij})$ и $B = (b_{ij})$, элементы которых суть значения выигрыша игроков на допустимых ситуациях. Отсюда следует еще одно часто используемое обозначение биматричной игры $\Gamma(A, B)$. Как обычно, игроки стремятся максимизировать свой выигрыш.

Ситуацией равновесия по Нэшу в смешанных стратегиях в биматричной игре $\Gamma(A, B)$ называется ситуация $(x^*, y^*) \in S_m \times S_n$, которая удовлетворяет неравенствам $H_1(x, y^*) \le H_1(x^*, y^*)$, $H_2(x^*, y) \le H_2(x^*, y^*)$ $\forall x \in S_m, y \in S_n$, где $S_m = \{x \in R^m: x_i \ge 0, \sum_i x_i = 1\}$, $S_n = \{y \in R^n: y_j \ge 0, \sum_j y_j = 1\}$ – стандартные симплексы; $H_1(x, y) = (xA, y)$, $H_2(x, y) = (xB, y)$ – выигрыши игроков в смешанном расширении игры $\Gamma(A, B)$. При этом стратегии $x^*$, $y^*$ называются равновесными.

Везде ниже, если не указано обратное, будем рассматривать только смешанное расширение биматричных игр, под термином "стратегия" будет пониматься смешанная стратегия, а под "ситуацией равновесия" – ситуация равновесия в смешанных стратегиях. Из доказанной Дж. Нэшем (Nash, 1950) теоремы следует, что в любой биматричной игре существует ситуация равновесия. Однако теорема Нэша указывает лишь на существование ситуации равновесия в биматричной игре, но не дает никакого алгоритма для ее нахождения. При этом в отличие от матричных игр, решение которых сводится к решению пары двойственных задач линейного программирования, решение биматричных игр приводит к невыпуклой оптимизационной задаче и, следовательно, является существенно более сложным.

## 2. ОПИСАНИЕ ДВОЙСТВЕННОГО МЕТОДА

Пусть дана биматричная игра с матрицами $A, B \subset R^{m\times n}$. Известно (Mills, 1960; Стрекаловский, Орлов, 2007), что ее решение сводится к решению билинейной задачи оптимизации:

$$\begin{cases} x(A+B)y - \alpha - \beta \to \max(=0); \\ xB \le \beta e_n; \\ Ay \le e_m\alpha; \\ xe_m = 1; \\ e_n y = 1; \\ x, y \ge 0. \end{cases} \quad (1)$$

Здесь через $e_m$, $e_n$ обозначаются векторы соответствующих размерностей, все координаты которых равны единице. Все векторы рассматриваются как строки или столбцы так, чтобы соответствующие матричные произведения имели смысл.

Если зафиксировать вектор $y$ и число $\alpha = \max_i (Ay)_i$, получим задачу линейного программирования $LP(y)$:

$$\begin{cases} x(A+B)y - \beta \to \max(\le \alpha); \\ xB - \beta e_n \le 0; \\ xe_m = 1; \\ x \ge 0. \end{cases} \qquad (2)$$

Очевидно, что задача (2) имеет решение при любом $y$ и ее значение непрерывно зависит от $y$. Составим двойственную к (2) задачу $DLP(y)$:

$$\begin{cases} t \to \min; \\ Bu + e_n t \ge (A+B)y; \\ -e_n u = -1; \\ u \ge 0. \end{cases} \qquad (3)$$

Очевидно, что вектор $u = y$ и $t = \alpha$ удовлетворяют ограничениям задачи (3). Следовательно, ее оптимальное значение не больше $\alpha$, причем равенство достигается тогда и только тогда, когда вектор $y$ является оптимальным. Обозначив множество оптимальных векторов задачи (3) через $u(y)$, получим, что решение биматричной игры сводится к нахождению неподвижной точки полунепрерывного сверху многозначного отображения $u(y)$ стандартного симплекса $S_n = \{y \in R^n_+, e_n y = 1\}$ в себя. Для поиска неподвижной точки можно использовать метод простой итерации. Однако задача (3) может иметь не единственное решение, поэтому в качестве критерия остановки лучше использовать не близость найденного решения к $y$, а значение целевой функции задачи (1). Поскольку теоретически обосновать сходимость этого метода не удалось, для исследования его эффективности был проведен ряд вычислительных экспериментов.

## 3. ОПИСАНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

**3.1. Основные результаты.** В проведенных экспериментах случайным образом генерировались матрицы $A$, $B$ размера от $10 \times 10$ до $100 \times 100$. В качестве начальных значений выбирались чистые стратегии второго игрока. Алгоритм заканчивал работу либо когда значение целевой функции задачи (1) при выбранном векторе $y$ и соответствующем ему решении задачи (2) $x$ оказывалось меньше заданной точности $\varepsilon$ (как правило, выбиралось значение $\varepsilon = 10^{-6}$), либо, если число сделанных итераций превышало заданное значение (1000). В последнем случае менялась начальная стратегия второго игрока и работа алгоритма возобновлялась. На рис. 1–5 показано распределение числа задач по числу использованных начальных стратегий.

Как видно из приведенных графиков, двойственный метод дает 100%-ную сходимость (за исключением матриц размера $10 \times 10$). Однако она достигается за счет выбора разных начальных стратегий. Так, например, для матриц $100 \times 100$ при выборе одной начальной стратегии результат достигается менее чем в половине случаев, а при выборе из нескольких альтернатив решение находится всегда. Доля решенных задач при выборе одной из четырех первых стратегий велика – не менее 75%, однако есть тенденция к снижению этого показателя с ростом размерности игры.

На рис. 6–8 показана зависимость количества и длительности итераций от размера задачи. Как видно из графиков, число итераций и время выполнения одной итерации не носят экспоненциальный характер, а скорее имеют вид линейных функций.

**3.2. Сравнение с алгоритмом Лемке–Хоусона.** Метод Лемке–Хоусона был исторически одним из первых подходов для решения биматричных игр. Он был разработан в начале 1960-х годов. Как показали исследования (McKelvey, McLennan, 1996), на сегодняшний день он яв-

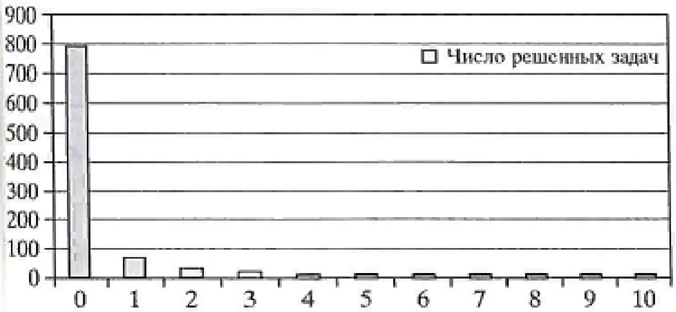


**Рис. 1.** Распределение числа задач по количеству использованных начальных стратегий для матриц 10 × 10

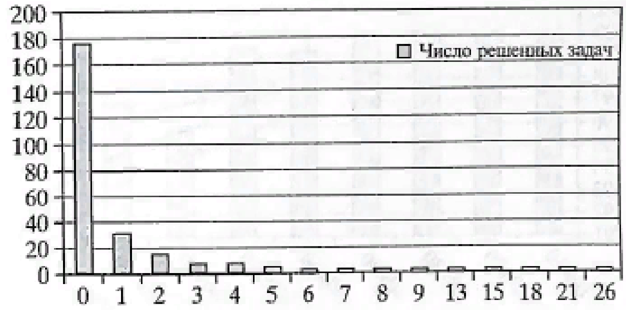


**Рис. 2.** Распределение числа задач по количеству использованных начальных стратегий для матриц 50 × 50

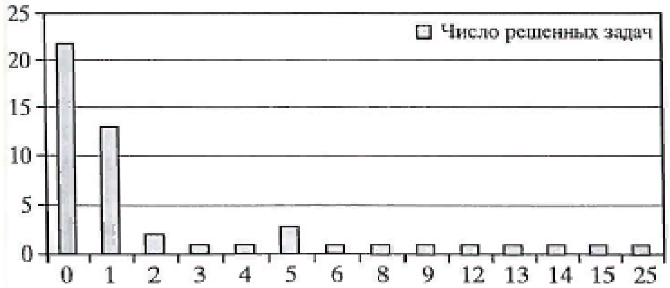


**Рис. 3.** Распределение числа задач по количеству использованных начальных стратегий для матриц 100 × 100

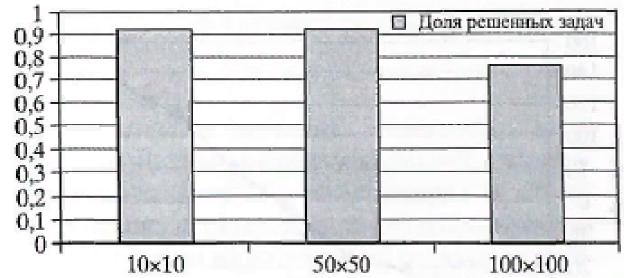


**Рис. 4.** Доля решенных задач при выборе одной из первых четырех стратегий

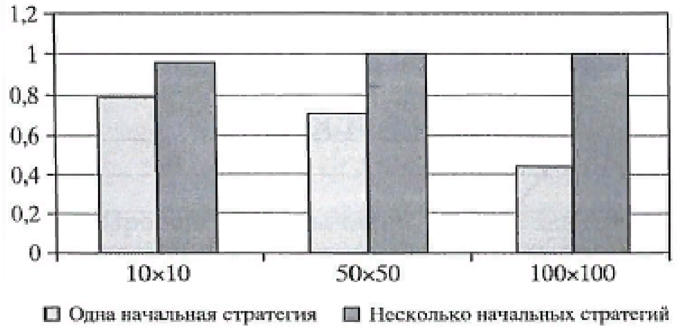


**Рис. 5.** Зависимость числа решенных задач от числа использованных начальных стратегий

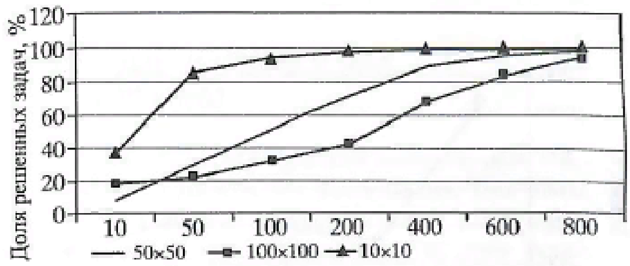


**Рис. 6.** Зависимость числа решенных задач от максимального числа итераций

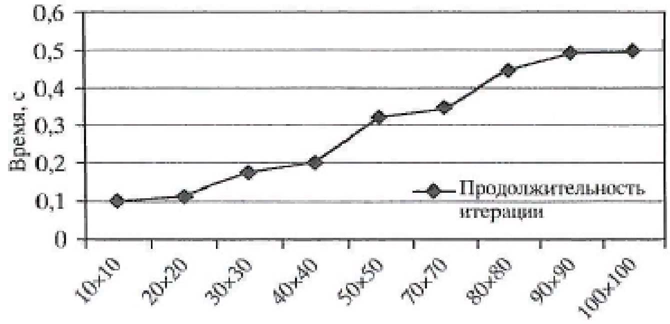


**Рис. 7.** Среднее время выполнения одной итерации

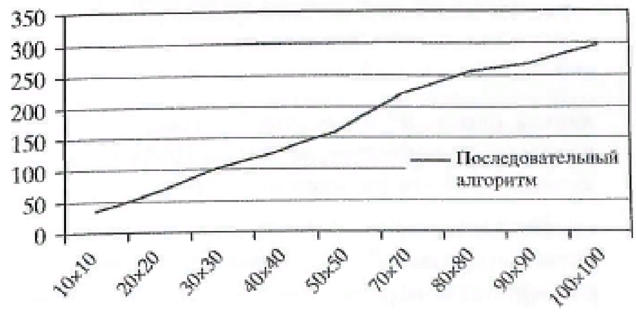


**Рис. 8.** Среднее число итераций


**Рис. 9.** Доля задач, решенных методом Лемке–Хоусона

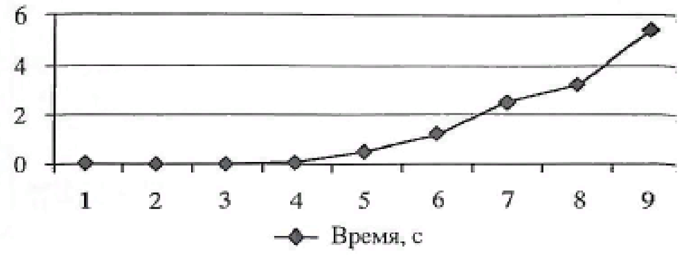


**Рис. 10.** Среднее время работы метода Лемке–Хоусона (ось абсцисс – размерность задачи, деленная на 10)


**Рис. 11.** Сравнительное время работы алгоритмов

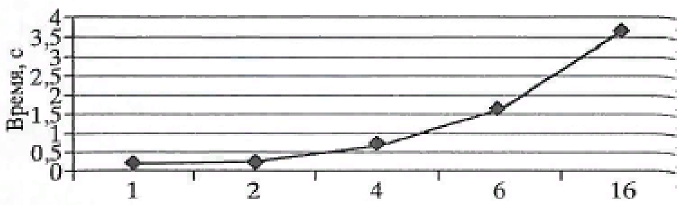


**Рис. 12.** Зависимость длительности итерации от числа потоков


**Рис. 13.** Зависимость среднего времени решения от числа потоков


**Рис. 14.** Зависимость количества итераций от числа потоков

ляется одним из наиболее быстрых алгоритмов отыскания равновесий в биматричной игре и показывает неплохую, хотя и не 100%-ную, сходимость. Есть тенденция к снижению сходимости метода с ростом размера игры (рис. 9).

На тестовом массиве данных время работы двойственного алгоритма уступает времени работы алгоритма Лемке–Хоусона, однако здесь требуется дополнительный анализ на больших размерностях игр, так как у второго алгоритма, возможно, есть тенденция к быстрому росту при увеличении размерности задач (рис. 10, 11).

**3.3. Параллельная модификация двойственного метода.** Так как на сходимость в значительной мере влияет выбор начального решения (стратегии второго игрока), то отсюда следует вывод:

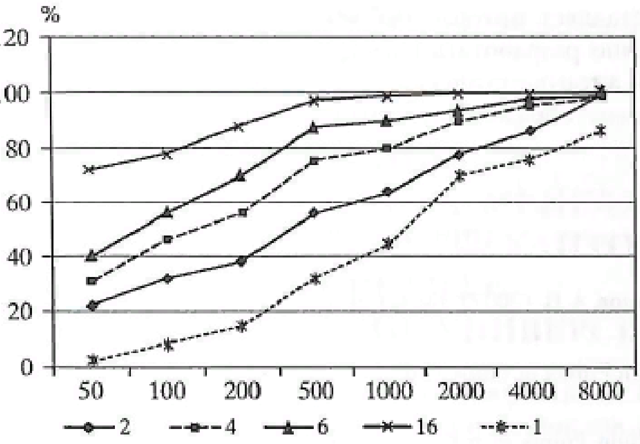


**Рис. 15.** Зависимость доли решенных задач от максимального числа итераций и числа потоков

*если искать решение, взяв сразу несколько стратегий в качестве начальных, можно сократить время поиска, увеличив вероятность нахождения решения.* Данный алгоритм позволяет достаточно простое распараллеливание: необходимо запустить некоторое число потоков, например, равное числу ядер процессора, в каждом из которых искать решение, отталкиваясь от разных начальных стратегий второго игрока. В случае если решение найдено одним из потоков, все остальные потоки завершаются; если же поток не нашел решения, то он выбирает следующую свободную чистую стратегию второго игрока (если такой нет, поток завершает свою работу).

В результате распараллеливания алгоритма время поиска решения существенно сократилось. При увеличении числа потоков увеличивается вероятность нахождения решения на меньшем номере итерации. Но увеличение числа потоков (больше числа ядер процессора) негативно сказывается на среднем времени продолжительности итерации (рис. 12–15).

## 4. ОСНОВНЫЕ ВЫВОДЫ И ВОЗМОЖНЫЕ НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ ИССЛЕДОВАНИЙ

Проведенные вычислительные эксперименты показали, что для случайно выбираемых матриц $A$, $B$ двойственный метод практически всегда позволяет найти точку равновесия. При этом, хотя по скорости работы он уступает алгоритму Лемке–Хоусона, этот разрыв имеет тенденцию к сокращению с ростом размерности задачи. Кроме того, при больших размерностях доля задач, решаемых двойственным методом, оказывается выше, чем у метода Лемке–Хоусона. Можно предположить, что использование более мощных компьютеров позволит повысить эффективность метода за счет распараллеливания и увеличения числа начальных стратегий.

Дальнейшие исследования могут быть направлены на выяснение следующих вопросов.

1. В проведенных экспериментах метод простой итерации применялся для поиска неподвижной точки функции, определенной на множестве стратегий второго игрока. Разумеется, можно подобрать аналогичную функцию и на множестве стратегий первого игрока. При этом есть возможность в процессе работы "переключаться" с одного игрока на другого. Не исключено, что чередование таких переключений окажется более эффективным, чем работа с одним и тем же игроком.

2. Может оказаться полезным сочетание двойственного метода с другими методами решения биматричных игр. Например, можно использовать полученные в ходе работы метода точки с достаточно большим значением целевой функции задачи (1) в качестве начальных для других методов.

3. Полученная в проведенных экспериментах сходимость двойственного метода может объясняться тем, что в играх с случайно выбираемыми матрицами число равновесий достаточно

велико. Поэтому представляет интерес работа метода в играх с малым числом равновесий. Впрочем, для этого нужно разработать алгоритм генерации таких игр. Возможно, этим свойством обладают “почти антагонистические” игры, у которых элементы матриц $A$ и $B$ упорядочены противоположным образом, или игры, матрицы которых содержат большое число нулевых элементов.

## СПИСОК ЛИТЕРАТУРЫ

**Стрекаловский А.С., Орлов А.В.** (2007). Биматричные игры и билинейное программирование. М.: Физматлит.

**Mills H.** (1960). Equilibrium Points in Finite Games // *J. of the Society for Industrial and Applied Mathematics.* Vol. 8.

**Nash J.F.** (1950). Equilibrium Points in n-Person Games. Proceedings of the National Academy of Sciences. Vol. 36.

**McKelvey R.D., McLennan A.** (1996). Computation of Equilibria in Finite Games. Handbook of Computational Economics. Vol. 1 / H.M. Amman, D.A. Kendrick, J. Rust (eds.). Vol. 1. Amsterdam, Holland: Elsevier.

Поступила в редакцию
24.02.2012 г.

## Dual Method for Solution of Bimatrix Games

**A.A. Zaslavsky, A.N. Lipatova**

The paper focuses on the research of the dual method for finding equilibrium points in bimatrix game. The convergence of this method isn't proved, but the computational experiments demonstrate that it can be used for solving the bimatrix games.

**Keywords:** bimatrix games, linear programming, dual problem.