ЭКОНОМИКА И МАТЕМАТИЧЕСКИЕ МЕТОДЫ, 2018, том 54, № 2, с. 1–15

МЕТОДЫ ОПТИМИЗАЦИИ

ГИБРИДНЫЙ МЕТОД ПОИСКА РЕШЕНИЯ БИМАТРИЧНЫХ ИГР

© 2018 г. Е. Г. Гольштейн[i], У. Х. Малков[ii], Н. А. Соколов[iii]

Аннотация. Для нахождения решения биматричной игры в смешанных стратегиях можно использовать приближенный метод решения биматричных игр (2LP-метод) и/или метод Лемке – Хаусона (LH-метод). В 2LP-методе поиск решения биматричной игры сводится к итеративному поиску глобального минимума функции Нэша, имеющего большое число локальных минимумов, не совпадающих с глобальным минимумом. Тем не менее поочередная минимизация этой функции по одной из двух переменных (стратегий) при фиксации другой переменной легко сводится к линейному программированию. Осуществляя перебор начальных чистых стратегий и решая на каждой итерации две задачи линейного программирования, 2LP-метод позволяет найти точное решение игры, если выполнено условие дополнительности либо некоторое приближение к множеству точек Нэша при незначительном нарушении условия дополнительности. Достоинством метода является его простота, главным недостатком – снижение эффективности при малой заполненности и/или при наличии взаимозависимости матриц, задающих функции выигрышей игроков. В LH-методе поиск решения биматричной игры заменяется поиском решения связанной с игрой системы линейных равенств. Начиная с единичного базиса метод делает шаги симплексного типа с целью уменьшить число нарушенных условий дополнительности. Как правило, но не всегда, этим методом удается найти точное решение игры. Предлагаемый нами гибридный метод производит дооптимизацию приближенного решения, полученного 2LP-алгоритмом, при помощи LH-алгоритма, использующего базис приближенного решения. Эффективность метода Лемке – Хаусона и нашего гибридного метода оказалась примерно одинаковой. С помощью гибридного метода удалось найти решение нескольких игр, для которых точное решение не было получено ни 2LP-методом, ни LH-методом.

Ключевые слова: биматричная игра, выпуклая структура, чистая стратегия, смешанная стратегия, точка Нэша, функция Нэша, условие дополнительности, метод Лемке–Хаусона, гибридный метод.

Классификация JEL: C02, C72.

DOI

Биматричная игра. Рассмотрим биматричную игру (т.е. конечную бескоалиционную игру двух лиц) Г, предполагая, что игроки 1 и 2 располагают m и n стратегиями соответственно. Пусть $A=(a_{ij})$ и $B=(b_{ij})$ – $(m\times n)$-матрицы, определяющие выигрыши игроков 1 и 2, использующих стратегии i $(1\leqslant i\leqslant m)$ и j $(1\leqslant j\leqslant n)$ соответственно.

Биматричная игра Г в смешанных стратегиях определяется двумя матрицами A и B и характеризуется множествами стратегий:

$$X=\left\{x=(x_1,\ldots,x_m)\in\mathbf{E}^m:\sum_{i=1}^{m}x_i=1,\quad x_i\geqslant 0,\quad 1\leqslant i\leqslant m\right\},$$
$$Y=\left\{y=(y_1,\ldots,y_n)\in\mathbf{E}^n:\sum_{j=1}^{n}y_j=1,\quad y_j\geqslant 0,\quad 1\leqslant j\leqslant n\right\} \tag{1}$$

и функциями выигрышей игроков

Grant

[i] **Евгений Григорьевич Гольштейн** – г.н.с., д.ф.-м.н., профессор, ЦЭМИ РАН, 117418, Москва, Нахимовский проспект, д. 47; golshtn@cemi.rssi.ru.

[ii] **Устав Херманович Малков** – в.н.с., к.ф.-м.н., ЦЭМИ РАН, 117418, Москва, Нахимовский проспект, д. 47; malkov@cemi.rssi.ru.

[iii] **Николай Александрович Соколов** – в.н.с., к.ф.-м.н., ЦЭМИ РАН, 117418, Москва, Нахимовский проспект, д. 47; sokolov@cemi.rssi.ru.

$$f_1(x,y) = \langle x, Ay \rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_i y_j, \quad f_2(x,y) = \langle x, By \rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}x_i y_j, \tag{2}$$

заданными на компакте $Z = X \times Y$ евклидова пространства $\mathbf{E} = \mathbf{E}^m \times \mathbf{E}^n$. Ввиду билинейности функций выигрышей $f_i(x,y)$ ($i = 1, 2$) Г – выпуклая игра, поэтому ее множество Z^* точек Нэша непусто, хотя и не обязано быть выпуклым множеством.

Определим на $z = (x,y) \in Z$ отображение $T_\Gamma(z) = (Ay, xB)$, порождающее вариационное неравенство $\langle T_\Gamma(z), z - z' \rangle \geqslant 0$ для всех $z' \in Z$. Будем говорить, что выпуклая игра Γ *имеет выпуклую структуру*, если отображение T_Γ монотонно. У всякой игры, имеющей выпуклую структуру, множество Z^* точек Нэша – непустой выпуклый компакт. Однако класс биматричных игр, имеющих выпуклую структуру, довольно узок: необходимым и достаточным условием для этого является соблюдение равенства $C = (c_{ij})_{mn} = A + B = (\mu_i + \nu_j)_{mn}$. В дальнейшем наличия у игры Г выпуклой структуры не предполагается.

Функция Нэша. Ввиду специфики компакта Z и функций выигрышей $f_1(z)$, $f_2(z)$ для любой точки $z = (x,y) \in Z$ имеем:

$$\begin{aligned} 0 &\leqslant \max_{x \in X} f_1(x,y) - f_1(z) \equiv \delta_1(z) \equiv \max_{1 \leqslant i \leqslant m} \sum_{j=1}^{n} a_{ij} y_j - f_1(z), \\ 0 &\leqslant \max_{y \in Y} f_2(x,y) - f_2(z) \equiv \delta_2(z) \equiv \max_{1 \leqslant j \leqslant n} \sum_{i=1}^{m} b_{ij} x_i - f_2(z). \end{aligned} \tag{3}$$

При помощи функций $\delta_q(z)$, $z \in Z$, $q = 1, 2$, заданных формулами (3), для игры Γ вводится стандартная (ненормированная) функция Нэша

$$F(z) = \delta_1(z) + \delta_2(z), \quad z \in Z. \tag{4}$$

Из (3), (4) и определения точки Нэша игры Γ следует, что $F(z) \geqslant 0$ для всех $z = (x,y) \in Z$ и, кроме того, $F(z) = 0$ в том и только в том случае, если z является точкой Нэша игры Γ. Значит, множество глобальных точек минимума функции $F(z)$ на Z совпадает с множеством Z^* точек Нэша игры Γ, причем значение минимума равно нулю. Таким образом, решение игры Γ оказывается эквивалентным поиску точек глобального минимума функции $F(z)$ на множестве Z, причем $F(z)$ можно считать естественной мерой отклонения точки z от множества Z^*.

2LP-метод решения биматричной игры (Гольштейн, Малков, Соколов, 2013). Хотя задача минимизации функции $F(z)$ по $z \in Z$ весьма сложна из-за наличия у нее многочисленных локальных минимумов, не совпадающих с глобальным, минимизация этой функции по одной из стратегий, составляющих вектор z, при фиксации другой стратегии легко сводится к линейному программированию.

Зафиксируем векторную переменную (стратегию) y значениями $y = y' = (y'_1, \ldots, y'_n) \in Y$ и рассмотрим задачу линейного программирования $P_1(y')$ с $m + n + 1$ переменными (x, v, α), где $v = (v_1, \ldots, v_n)$ – вектор дополнительных переменных, и $n + 1$ ограничениями-равенствами:

$$\begin{aligned} &\sum_{i=1}^{m}\Big(\sum_{j=1}^{n} - c_{ij} y'_j\Big) x_i + \alpha \to \min, \\ &\sum_{i=1}^{m} b_{ij} x_i + v_j = \alpha, \quad j = 1, \ldots, n, \\ &\sum_{i=1}^{m} x_i = 1, \quad x_i \geqslant 0, \quad i = 1, \ldots, m, \\ &v_j \geqslant 0, \quad j = 1, \ldots, n, \quad \alpha \in \mathbf{E}^1. \end{aligned} \tag{5}$$

Обозначим $x^* = x^*(y')$ первые m компонент оптимального плана задачи (5), тогда $F(x^*,y')\leqslant F(x',y')$ для любой точки $x' \in X$.

Аналогичным образом, зафиксировав другую стратегию x значениями $x = x' \in X$, введем задачу линейного программирования $P_2(x')$ с $n+m+1$ переменными (y,u,β) $(u = (u_1,\ldots,u_m)$ – вектор дополнительных переменных) и $m+1$ ограничениями-равенствами:

$$\begin{gathered}\sum_{j=1}^{n}(\sum_{i=1}^{m} - c_{ij}x'_i)y_j + \beta \to \min,\\ \sum_{j=1}^{n} a_{ij}y_j + u_i = \beta, \quad i = 1,\ldots,m,\\ \sum_{j=1}^{n} y_j = 1, \quad y_j \geqslant 0, \quad j = 1,\ldots,n,\\ u_i \geqslant 0, \quad i = 1,\ldots,m, \quad \beta \in \mathbf{E}^1.\end{gathered} \tag{6}$$

Пусть $y^* = y^*(x')$ – первые n компонент оптимального плана задачи $P_2(x')$. Очевидно, что $F(x',y^*)\leqslant F(x',y')$ для произвольной точки $y' \in Y$.

Пусть Z' – множество точек $z^* \in Z$, являющихся точками локального минимума функции $F(z)$ на Z , а множество Z'' состоит из точек $z^* = (x^*,y^*) \in Z$, таких что x^* – решение задачи $P_1(y^*)$, y^* – решение задачи $P_2(x^*)$. Имеет место включение $Z^* \subset Z' \subset Z''$, где Z^* – введенное ранее множество точек Нэша игры Γ, или множество точек глобального минимума функции $F(z)$ на Z . Кроме того, можно показать, что любая точка $z^* = (x^*,y^*)$, принадлежащая множеству Z'' , содержится в множестве Z' , если хотя бы одна из задач $P_1(y^*)$, $P_2(x^*)$ имеет единственное решение. Поэтому точки из множества $Z'' \setminus Z'$ попадаются довольно редко.

Отправляясь от произвольной *стартовой* стратегии $y^0 \in Y$ игрока 2, построим последовательность точек $\{z^t\} = \{(x^t,y^t)\}$, $t\geqslant 1$, следующим образом: x^{t+1} – решение задачи $P_1(y^t)$, y^{t+1} – решение задачи $P_2(x^{t+1})$, $t = 0,\ 1,\ldots$.

По аналогичным правилам можно определить последовательность $\{z^t\}$, $t\geqslant 1$, отправляясь от произвольной стартовой стратегии $x^0 \in X$ игрока 1: y^{t+1} – решение задачи $P_2(x^t)$, x^{t+1} – решение задач и $P_1(y^{t+1})$, $t = 0,\ 1,\ldots$.

Ясно, что в обоих случаях $F(z^{t+1})\leqslant F(z^t)$ для всех $t\geqslant 1$, поэтому существует $Q = Q(z^0) = \lim_{t\to\infty} F(z^t)\geqslant 0$. Множество предельных точек последовательности $\{z^t\}$ непусто и содержится в Z'' , значения функции Нэша во всех предельных точках одинаковы и равны $Q(z^0)$.

Критерий Нэша (окончания счета), условие дополнительности. Пусть $\overline{X} = \{x_1,\ldots,x_r\}$ $(r\geqslant 0)$ и $\overline{Y} = \{y_1,\ldots,y_s\}$ $(s\geqslant 0)$ – заранее заданные множества стартовых стратегий игроков 1 и 2 соответственно. Обычно эти множества включают в себя множества чистых стратегий (полностью или частично), а также, возможно, некоторое количество сгенерированных смешанных стратегий. Для сокращения вычислительных усилий, упрощения алгоритма и уменьшения времени счета возможен случай $\overline{X} = \varnothing$ или $\overline{Y} = \varnothing$, где $rs = 0$, $r + s > 0$. Пусть $\overline{Z} = \{z_1,\ldots,z_{r+s}\} = \overline{X} \cup \overline{Y}$, причем элементы этого множества могут быть перетасованы; $S = |\overline{Z}|$ – число стартовых стратегий.

Определим число $\overline{Q}(\overline{Z}) = \min\{Q(z_k) : 1\leqslant k\leqslant r+s\}$ – наименьшее значение функции Нэша $F(z)$, которое можно получить при использовании множества $\overline{Z}$ всех стартовых стратегий. Разумеется, $\overline{Q}(\overline{Z})$ не обязано быть равным нулю.

Метод становится приближенным, если для каждой стартовой стратегии $z \in \overline{Z}$ процесс отыскания $\lim_{t\to\infty} F(z^t)$ заменить на поиск $t(z) = \min\{t \in \{1,\ldots,\overline{T}\} : F(z^t) - F(z^{t-1})\leqslant \varepsilon_F\}$, где ε_F – заранее заданное небольшое положительное число, $\overline{T}$ – наибольшее разрешенное число итераций (пар задач вида (5), (6)). Положим $t(z) = \overline{T}$ при невыполнении неравенства для всех t . Если обозначить $\widetilde{Q}(z) = F(z^{t(z)})$, то величина $\widetilde{Q}(\overline{Z}) = \min_{z\in\overline{Z}} \widetilde{Q}(z) \geqslant \overline{Q}(\overline{Z}) \geqslant 0$.

В качестве критерия окончания счета можно взять выполнение условия

$$F(z) \leqslant \varepsilon_f, \tag{7}$$

где ε_f – заранее заданное небольшое положительное число.

Согласно (3), (5) и (6) для любой стратегии $z \in Z$ справедливо представление

$$\delta_1(z) = \sum_{i=1}^{m} x_i u_i, \quad \delta_2(z) = \sum_{j=1}^{n} y_j v_j.$$

Поэтому для функции Нэша $F(z)$ имеет место и такое представление:

$$F(z) = \sum_{i=1}^{m} x_i u_i + \sum_{j=1}^{n} y_j v_j. \tag{8}$$

Пары переменных x_i и u_i, y_j и v_j назовем *взаимодополнительными*. Поскольку все переменные в формуле (8) неотрицательны, для справедливости включения $z \in Z^*$ необходимо и достаточно, чтобы в каждой паре взаимодополнительных переменных хотя бы один из сомножителей был равен нулю. При выполнении этого условия будем говорить, что выполняются *все условия (взаимо) дополнительности* (или имеет место *условие дополнительности*).

Пусть $\mathbf{I}\{A\}$ – индикатор события $\{A\}$, принимающий значение 1, если событие A истинно, и 0, если событие A ложно. Функция

$$\Delta_\varepsilon = \Delta_\varepsilon(z,u,v) = \sum_{i=1}^{m} \mathbf{I}\{x_i > \varepsilon\}\mathbf{I}\{u_i > \varepsilon\} + \sum_{j=1}^{n} \mathbf{I}\{y_j > \varepsilon\}\mathbf{I}\{v_j > \varepsilon\} \tag{9}$$

есть число нарушенных в точке $z \in Z$ условий взаимодополнительности. Здесь ε – небольшая неотрицательная константа: числа, меньшие ε, в формуле (9) считаются нулями.

Проведенное тестирование (Гольштейн, Малков, Соколов, 2013) показало, что 2LP-метод отыскивает точное решение игры, если выполнено условие дополнительности либо некоторое приближение к множеству точек Нэша при незначительном нарушении условия дополнительности. Достоинством метода является его простота, а главным недостатком – снижение эффективности при малой заполненности и/или при наличии взаимозависимости матриц, задающих функции выигрышей игроков.

Метод Лемке – Хаусона решения биматричных игр (Lemke, Howson, 1964). В LH-методе поиск решения биматричной игры заменяется на поиск решения связанной с игрой системы линейных равенств линейной задачи дополнительности.

Сделав в (5)—(6) замену переменных $x := x / \alpha$, $y := y / \beta$, получим систему линейных равенств и неравенств (назовем ее *LH-задачей*):

$$\begin{gathered}
\sum_{j=1}^{n} a_{ij} y_j + u_i = 1, \quad i = 1,\dots,m, \\
\sum_{i=1}^{m} b_{ij} x_i + v_j = 1, \quad j = 1,\dots,n, \\
x_i \geqslant 0, \quad u_i \geqslant 0, \quad i = 1,\dots,m, \\
y_j \geqslant 0, \quad v_j \geqslant 0, \quad j = 1,\dots,n,
\end{gathered} \tag{10}$$

в которой отсутствуют равенства $\sum_{i=1}^{m} x_i = 1$, $\sum_{j=1}^{n} y_j = 1$. Введя обозначения

$$H = \begin{pmatrix} O_m & A \\ B^{\mathrm{T}} & O_n \end{pmatrix}, \quad \begin{matrix} z = (z_1,\dots,z_{m+n}) \equiv (x,y), \\ w = (w_1,\dots,w_{m+n}) \equiv (u,v), \end{matrix} \quad \begin{matrix} e = (1,1,\dots,1) \in \mathbf{E}^{m+n}, \\ o = (0,0,\dots,0) \in \mathbf{E}^{m+n}, \end{matrix} \quad E = \mathrm{diag}(e),$$

где O_m и O_n – квадратные матрицы размера m и n соответственно, все элементы у которых нулевые, E – единичная матрица размера $m+n$, символ T означает транспонирование матрицы, приходим к *линейной задаче дополнительности*: найти неотрицательное ($z \geqslant 0$, $w \geqslant 0$) решение линейной системы

$$Hz + Ew = e, \tag{11}$$

удовлетворяющее условию дополнительности $\Delta_0(z,w) = 0$, т.е. $z_i w_i = 0$ для всех $1 \leqslant i \leqslant m+n$. Будем говорить, что решение (z,w) системы (11) *почти* удовлетворяет условиям дополнительности, если $\Delta_0(z,w) = 1$, т.е. $z_i w_i \neq 0$ только для одного значения индекса i, $1 \leqslant i \leqslant m+n$.

В невырожденном случае множество вершин выпуклого многогранного множества $\mathbf{M} = \{(z,w) \in \mathbf{E}^{2(m+n)} : \ Hz + Ew = e, \ \ z = o, \ \ w \geqslant o\}$ находится во взаимнооднозначном соответствии с множеством невырожденных допустимых базисных решений системы (11).

Начальный базис LH-задачи составлен из столбцов матрицы E, соответствующих дополнительным переменным w. Для него выполняется условие дополнительности, но $z = o$, т.е. не выполнены условия $\sum_{i=1}^{m} x_i = 1$, $\sum_{j=1}^{n} y_j = 1$. Введя в базис одну из структурных переменных (например, x_1 – первую компоненту вектора z^1) и тем самым нарушив одно условие дополнительности ($x_1 u_1 = 0$), LH-метод осуществляет далее шаги (итерации) симплексного типа с целью добиться выполнения нарушенного условия дополнительности. Генерируемая последовательность вершин, имеющая началом точку (z^1, w^1), почти удовлетворяющих условиям дополнительности, образует так называемый *путь Лемке* по допустимым вершинам множества $\mathbf{M}$. После нескольких итераций LH-метода у структурных переменных x и y появится несколько ненулевых компонент. Можно сделать обратное преобразование переменных к первоначальному виду и в дальнейшем выполнять шаги (путь) Лемке уже в базисе исходной задачи с учетом $\sum_{i=1}^{m} x_i = 1$, $\sum_{j=1}^{n} y_j = 1$. В этом случае исключается возможность возврата в исходную точку $z = o$.

Как правило, но не всегда, методу удается найти точное решение линейной системы (11), а значит, и игры Γ. Чтобы вернуться обратно к исходным переменным, надо провести нормировку полученного решения системы: $x := x \,/ \sum_{i=1}^{m} x_i$, $y := y \,/ \sum_{j=1}^{n} y_j$.

Каждая итерация LH-метода соответствует (как и в обычном симплекс-методе) движению из вершины (z^l, w^l), $l \geqslant 1$, множества $\mathbf{M}$, вдоль некоторого его ребра, почти удовлетворяющего условиям дополнительности. Такое движение заканчивается в смежной вершине (z^{l+1}, w^{l+1}), в которой $\Delta_0(z^{l+1}, w^{l+1}) \leqslant 1$. Процесс прекращается, если текущая точка уже генерировалась ранее (получен цикл) или она удовлетворяет условиям дополнительности точно, т.е. является решением линейной системы.

Основу LH-метода (и его производных алгоритмов LH1 и LH2) составляет *LH- процедура* – такая последовательность замен базиса, для которой выполняется **главное правило**: выводимая из базиса переменная выбирается из условия сохранения неотрицательных значений базисных переменных, а в текущий базис вводится та неизвестная (z_k или w_k, $1 \leqslant k \leqslant n+m$), которая является дополнительной к неизвестной (соответственно w_k или z_k), покинувшей базис на предыдущем шаге (итерации).

Таким образом, LH-процедура определяет однозначно путь Лемке после выбора стартовой переменной для ввода в начальный базис, составленный из дополнительных переменных (при условии невырожденной игры двух лиц). Правило вывода переменной из базиса и формулы, его реализующие, те же, что имеют место в симплекс-методе.

Недостатком LH-метода является возможность зацикливания (возврат к исходной точке) либо недопустимо большое число итераций. Не получив решения при выборе $z = z_1$, начинаем процесс заново при $z = z_2$, и т.д. Считаем игру нерешенной, если опробованы все точки z_i, $i = 1,\dots,n+m$, а решение не найдено.

Тестирование показало, что имеет смысл отбраковывать точки z_i, $i = 1,\dots,m+n$, пытаясь делать для каждой не более $I(n+m)$ итераций при $0{,}01 < I \leqslant 5$. Эффективность метода существенно зависит от значения константы I, которую желательно подбирать для каждой игры.

LH1-алгоритм с горячим стартом. Точку $z = (x,y) \in \mathbf{E}^{m+n}$ будем считать приближенным решением биматричной игры Γ, если $F(z) \leqslant \varepsilon_f$ и $\Delta_\varepsilon \leqslant 1$, для точного решения игры Γ нужно, чтобы $F(z) \ll \varepsilon_f$ и $\Delta_\varepsilon = 0$; функции $F(z)$ и $\Delta_\varepsilon(z,w)$ определены в (7)–(9). Назовем *перспективной* точку $z = (x,y)$, для которой $0 < \varepsilon_f < F(z) \leqslant \overline{F}$ и $1 \leqslant \Delta_\varepsilon(z,w) \leqslant \overline{\Delta}$, где $\overline{F}$ и $\overline{\Delta}$ – некоторые специально подобранные константы.

Пусть z – перспективная стартовая точка игры Γ. Применим к ней вариант LH-метода, целью которого является отыскание решения (точки Нэша) с меньшим количеством нарушенных условий дополнительности. Этот вариант будем называть *LH1- алгоритмом*.

В отличие от LH-метода (с холодным стартом), в котором в качестве начального базиса задается единичная матрица, а условие дополнительности $\Delta_0(z,w) = 0$ выполняется при $z = o$, $w = e$, в LH1-алгоритме $\Delta_0(z,w) = k > 0$, т.е. в начальном базисе содержатся (в произвольном порядке) $2k$ базисных переменных z_i, w_i, $i = i_1,\ldots,i_k$ (назовем их нарушителями), для которых $z_i w_i > 0$. Одновременно имеется $2k$ небазисных переменных (кандидатов для ввода в базис), для которых $z_j = w_j = 0$, $j = j_1,\ldots,j_k$. Чтобы уменьшить число нарушенных условий дополнительности, нужно ввести в базис некоторые переменные-кандидаты вместо нарушителей.

Начальный базис LH-задачи для горячего старта составляется из базисов задач (5) и (6) (кроме переменных α и β), полученных после применения 2LP-метода. Для определения, в какие позиции вводятся в базис переменные из списка составленного начального базиса, вычислим разложение по текущему базису столбца, соответствующего очередному кандидату из этого списка. В качестве позиции для ввода в базис выбираем максимальный по модулю коэффициент этого разложения среди еще незанятых позиций. Пусть z – перспективная стартовая точка игры Γ. В качестве позиции для ввода в базис выбираем максимальный по модулю коэффициент этого разложения среди незанятых позиций. Вводим очередного кандидата в базис в выбранную позицию и повторяем эту процедуру до просмотра полного списка составленного начального базиса.

Попробуем последовательно ввести в базис переменные-кандидаты вместо нарушителей. Если это получится и все условия дополнительности будут выполнены, то LH-задача (10) решена. А если при попытке ввода в базис очередного кандидата будет выведен из базиса не нарушитель, а переменная, не входящая в нарушенное условие дополнительности, то при помощи LH-процедуры вводим в базис ту переменную, которая связана с вышедшей из базиса переменной по условию дополнительности; повторяем эту операцию, пока либо удастся вытеснить из базиса нарушителя и Δ_0 уменьшится, либо число этих попыток не достигнет предельного уровня $\overline{I}m$, где $\overline{I}$ – заданная константа; возможно, процесс зациклится до достижения этого предельного значения. При этом происходит переход к следующему кандидату без изменения Δ_0.

В проведенном вычислительном эксперименте в большинстве случаев предложенным способом удалось из всех пар нарушителей вывести из базиса по одной переменной из каждой пары, т.е. убрать все нарушенные условия дополнительности.

LH2-алгоритм с дополнительным столбцом. Если не удалось при помощи LH1-алгоритма вывести из базиса всех нарушителей, попытаемся устранить их, построив так называемый *дополнительный столбец.*

Сначала нужно принудительно заменить одного нарушителя в каждой паре на одного кандидата из пар кандидатов для устранения $\overline{k}$ оставшихся после LH1-алгоритма нарушенных условий дополнительности. Пусть $k_x \geqslant 0$ и $k_y \geqslant 0$ – число нарушенных условий дополнительности, относящихся к переменным x и y соответственно.

При $k_x > 0$ среди всех возможных k_x^2 замен для ввода в базис выбираем кандидата, максимизирующего минимальное отрицательное значение базисных переменных, которые наверняка появятся при такой принудительной замене базиса. Уменьшаем k_x на 1, и процесс повторяется, пока не получим $k_x = 0$. Аналогичные действия производим при $k_y > 0$.

Выбрав очередного кандидата, вычисляем разложение по базису столбца, соответствующего выбранному кандидату. Среди коэффициентов этого разложения найдем те, которые находятся в позициях, соответствующих нарушителям. Среди них определим максимальный или достаточно большой по абсолютному значению, и в эту позицию вводим в базис выбранного кандидата.

Дополнительный столбец надо построить таким образом, чтобы после ввода его в базис исчезли отрицательные значения базисных переменных. Для этого дополнительный столбец собираем как сумму столбцов, которым соответствуют переменные, имеющие отрицательные значения в базисном решении. Этот столбец с весом минимального значения (максимального по абсолютному значению среди отрицательных) среди базисных переменных вводим в базис в позицию, где в базисе была переменная, имеющая это значение. В итоге все значения базисных переменных станут неотрицательными, а переменная, соответствующая дополнительному столбцу, будет иметь значение, равное единице.

Далее запускается LH-процедура, в качестве начальной точки для ввода в базис выбираем переменную, связанную по условию дополнительности с переменной, вышедшей из базиса при вводе туда дополнительного столбца. Повторяем шаги LH-процедуры до выхода из базиса дополнительного столбца, т.е. до получения точки Нэша, или до достижения заданного предельного количества итераций.

Гибридный метод решения биматричной игры. Теперь, когда определены составляющие части (2LP-метод, LH1- и LH2-алгоритмы), опишем работу гибридного метода отыскания решения биматричной игры как альтернативы LH-метода и как возможного способа нахождения решения игры в случае когда LH-метод не может отыскать точное решение игры.

Шаг 0. Задаем константы, фиксируем множество стартовых точек.

Шаг 1. Выбераем очередную стартовую точку из множества стартовых точек, отправляясь от которой, запускаем 2LP-метод, решая поочередно задачи линейного программирования, пока процесс минимизации функции Нэша не стабилизируется.

Шаг 2. Если значение функции Нэша достаточно мало и (почти) выполнены условия дополнительности, т.е. $F(z) \leqslant \varepsilon_f$ и $\Delta_\varepsilon(z,w) \leqslant \underline{1}$, то игра Γ считается решенной с достаточно высокой точностью. Если значение $F(z) > \overline{F}$ или $\Delta_\varepsilon(z,w) > \overline{\Delta}$, то при наличии неопробованных стартовых точек переходим к шагу 1 с очередной стартовой точкой. Если же все множество стартовых точек исчерпано, констатируем, что решение игры Γ не найдено.

Шаг 3. Стартовая точка является перспективной. Используя при помощи 2LP-алгоритма полученное решение, составим базис задачи LH. Найдем нарушенные условия дополнительности и составим списки нарушителей и кандидатов.

Шаг 4. Просматривая список пар кандидатов при помощи LH1-алгоритма, вводим в базис одного из кандидатов из каждой пары. Если на очередном шаге этого просмотра удалось вывести из базиса нарушителя, переходим к следующей паре кандидатов с новым базисом. В противном случае продолжим со старым базисом. Если в итоге удалось вывести из базиса всех нарушителей, точка Нэша получена и игра решена. Если нет, то переходим к построению дополнительного столбца.

Шаг 5. Строим дополнительный столбец и применяем LH2-алгоритм. Если получена точка Нэша, то игра решена. В противном случае переходим к шагу 1 для следующей стартовой точки, если множество стартовых точек еще не исчерпано, и поиск решения игры продолжается заново. В противном случае решение игры не найдено.

Вычислительный эксперимент. Тестируемые методы (2LP, LH и гибридный) реализованы средствами пакета MATLAB7.14.0.739 R2012a (MATLAB (2012)). Подзадачи линейного программирования решались при помощи процедуры cplexlp пакета IBM ILOG CPLEX 12.6.2 (IBM ILOG CPLEX (2011)), значительно превосходящей процедуры linprog пакета MATLAB. Для большинства расчетов был использован персональный компьютер IntelCore i5–2400 CPU (3.1 GHz), 4 Gb RAM.

Для тестирования и сравнения эффективности методов в качестве основного использовалось семейство биматричных игр, содержащее 15 серий игр среднего размера, задаваемых параметрами $m, n \in \{20, 40, 60, 80, 100\}$, $m \leqslant n$. Для каждой серии вначале генерировалось по $\overline{R} = 100$ задач. Если не оговорено особо, то основные параметры метода таковы: $S = m$, $T = 90$, $\varepsilon_F = 10^{-8}$, $\varepsilon_f = 10^{-6}$, $\varepsilon = 10^{-4}$, $\overline{F} = 0{,}01$, $I = \overline{\Delta} = 5$, $\overline{I} = 1$.

Для экономии вычислительных усилий и времени счета в качестве множества стартовых точек взято упорядоченное множество чистых стратегий $X^0 = \{e_m^1,\ldots,e_m^m\}$ игрока 1. Здесь e_m^i – вектор из $\mathbf{E}^m$, у которого все координаты, кроме координаты i, равны нулю, а на месте i стоит единица.

Генерация матриц $A = (a_{ij})$ и $B = (b_{ij})$ осуществлялась в два этапа. Сначала при помощи датчика псевдослучайных чисел пакета MATLAB, равномерно распределенных в интервале (0, 1), генерировались независимо числа a'_{ij} и b'_{ij}. Затем осуществлялось преобразование $a_{ij} = a'_{ij} + \varkappa b'_{ij},\ \ b_{ij} = b'_{ij} + \varkappa a'_{ij},\ \ 1\leqslant i\leqslant m$, $1\leqslant j\leqslant n$. Здесь $\varkappa$ – *коэффициент взаимозависимости матриц*, $0\leqslant\varkappa\leqslant 1$. Поэтому элементы матриц находятся в диапазоне $(0,1+\varkappa)$.

При проведении вычислительных экспериментов исследовалось поведение и проводилось сравнение эффективности 2LP-, LH- и гибридного методов решения биматричных игр. Также нас интересовал вопрос, можно ли, использовав 2LP-метод, получить такую начальную точку, отправляясь от которой при помощи LH-метода мы отыскали бы решение игры, затратив меньшее количество вычислительных усилий (итераций, времени и т.д.).

Мы не будем останавливаться на вычислительном опыте решения биматричных игр при помощи 2LP-метода, поскольку подобный опыт (для близкой реализации) уже описан нами в работе (Гольштейн, Малков, Соколов, 2013). Напомним, что метод оказался эффективным для независимых и слабо зависимых матриц A и B, но с ростом параметра взаимозависимости (при $\varkappa\geqslant 0,2$) эффективность 2LP-метода падает.

Результаты тестирования семейства, содержащего 1500 игр, при помощи LH-метода для трех значений параметра взаимозависимости ($\varkappa \in \{0;\ 0,1;\ 0,2\}$) приведены в табл. 1, где p – номер серии, $1\leqslant p\leqslant 15$; m,n – размеры биматричной игры; параметр I задает предельно допустимое число (равное $I(m+n)$) итераций в LH-методе; множество $R = R^{(p)} = \{1,\ldots,\overline{R}\}$ (индекс p далее опущен) разбито на три непересекающихся (возможно, пустых) подмножества: $R = R_{PS} \cup R_{LH} \cup R_{NO}$; $k_{PS} = |R_{PS}|$ – число игр, у которых имеются решения в чистых стратегиях и которые не нужно решать LH-методом; $k_{LH} = |R_{LH}|$ – число игр, решенных LH-методом; $k_{NO} = |R_{NO}|$ – число игр, которые не удалось решить LH-методом (итого: $k_{PS} + k_{LH} + k_{NO} = \overline{R} = 100$ игр); $\rho = R \setminus R_{PS}$ – множество игр, решаемых LH-методом. Далее, s_π, j_π и t_π ($\pi\in\rho$) – суммарное число стартовых точек, число итераций и время, соответственно, понадобившиеся LH-методу для нахождения решения k_{LH} серии игр p и/или для установления факта, что решения k_{NO} игр серии p не найдены, при этом для каждой нерешенной игры число стартовых точек равно $m+n$ и число итераций для каждой стартовой точки из $\pi\in R_{\mathrm{NO}}$ равно $I(m+n)$; величины S_{sum} – общее число стартовых точек, $J_{\mathrm{L}H}$ – общее число итераций симплексного типа, *Time* – время (в секундах), потребовавшееся LH-методу для поиска решения, S_{max} – максимальное число стартовых точек на одну решаемую игру, при этом выполняются следующие формулы

$$S_{sum} = \sum_{\pi\in\rho} s_\pi,\quad S_{max} = \max_{\pi\in\rho} s_\pi,\quad J_{LH} = \sum_{\pi\in\rho} j_\pi,\quad Time = \sum_{\pi\in R} t_\pi. \tag{12}$$

Анализ данных, представленных в табл. 1, позволяет сделать следующие выводы.

1. Генерация независимых матриц A и B приводит к получению 935 игр, имеющих решения в чистых стратегиях. Следовательно, при $\varkappa = 0$ решать пришлось лишь 565 игр. При $\varkappa = 0,1$ пришлось решать 1477 игр, при $\varkappa = 0,2$ – 1496 игр.

2. При $\varkappa\leqslant 0,1$ были решены LH-методом ВСЕ игры семейства, при $\varkappa = 0,2$ не удалось решить 25 игр из 1500.

3. Примерно такие же результаты были получены при тестировании приближенного метода решения биматричных игр (Гольштейн, Малков, Соколов, 2013). Мы не будем количественно сравнивать 2LP-метод с LH-методом, поскольку имеются небольшие отличия при генерации матриц A и B, расчеты проводились на разных персональных компьютерах с различными ресурсами, использовано различное программное обеспечение и т.д.

Таблица 1. Зависимость от параметра $\varkappa$ результатов решения LH-методом тестового набора $\overline{R} = 100$ игр

$\varkappa$	I	№ п/п	Размеры		Результат			Общее число			*Time*, с
			m	h	k_{PS}	k_{LH}	k_{NO}	S_{sum}	S_{max}	J_{LH}	
0	0,5	1	20	20	63	37	0	68	6	1047	1,27
		2	20	40	66	34	0	81	7	1906	2,57
		3	20	60	66	34	0	66	5	1820	3,07
		4	20	80	60	40	0	66	7	2140	3,91
		5	20	100	71	29	0	54	6	2103	4,03
		6	40	40	61	39	0	87	8	2704	4,53
		7	40	60	57	43	0	103	7	3972	6,55
		8	40	80	66	34	0	62	5	2642	4,57
		9	40	100	63	37	0	78	8	3933	7,42
		10	60	60	67	33	0	115	23	5837	9,77
		11	60	80	55	45	0	133	15	7603	14,35
		12	60	100	59	41	0	116	22	7429	14,93
		13	80	80	65	35	0	99	10	6125	12,60
		14	80	100	60	40	0	94	6	6220	14,32
		15	100	100	56	44	0	142	10	11242	25,16
		Сумма			935	565	0	1364	—	66723	129,05
0,1	5	1	20	20	12	88	0	88	1	3829	3,11
		2	20	40	3	97	0	97	2	6636	6,61
		3	20	60	6	94	0	99	3	12047	12,92
		4	20	80	0	100	0	107	2	16709	19,56
		5	20	100	0	100	0	108	3	22839	30,12
		6	40	40	0	100	0	140	8	28901	28,64
		7	40	60	0	100	0	143	4	38481	43,74
		8	40	80	0	100	0	196	8	79295	101,90
		9	40	100	2	98	0	169	8	75281	110,40
		10	60	60	0	100	0	218	9	96196	124,00
		11	60	80	0	100	0	256	9	137887	198,85
		12	60	100	0	100	0	305	17	200335	324,17
		13	80	80	0	100	0	345	18	232845	376,62
		14	80	100	0	100	0	414	20	320417	579,53
		15	100	100	0	100	0	525	22	473061	935,22
		Сумма			23	1477	0	3210	—	1744759	2894,36
0,2	5	1	20	20	4	96	0	97	2	5521	4,25
		2	20	40	0	100	0	105	3	11251	8,63
		3	20	60	0	100	0	109	3	17419	16,59
		4	20	80	0	100	0	118	3	27961	29,85
		5	20	100	0	100	0	116	3	35559	43,03
		6	40	40	0	100	0	158	9	42513	171,31
		7	40	60	0	100	0	245	11	99781	108,75
		8	40	80	0	100	0	320	10	162693	204,61
		9	40	100	0	100	0	366	15	225518	319,31
		10	60	60	0	100	0	510	25	279997	348,93
		11	60	80	0	100	0	795	53	527947	744,45
		12	60	100	0	99	1	1153	60	891483	1405,23
		13	80	80	0	100	0	1358	61	1050087	1655,51
		14	80	100	0	92	8	2335	80	2070447	3862,36
		15	100	100	0	84	16	4075	100	4039640	7782,42
		Сумма			4	1471	25	11860	—	9487817	16705,23

4. На эффективность работы LH-метода существенно влияет подбор параметра I, задающего максимально разрешенное число итераций при решении одной игры. При подборе подходящего значения I следует учитывать величину параметра $\varkappa$ и размеры решаемой задачи. Для наших расчетов подходящими (но, возможно, неоптимальными) оказались значения $I = 0{,}5$ при $\varkappa = 0$ и $I = 5$ при $\varkappa$, равном 0,1 или 0,2.

5. При изменении параметра $\varkappa$ от 0 до 0,1 среднее значение $\tilde{S}$ числа стартовых точек на одну решаемую задачу меняется незначительно ($\tilde{S} = 1364 / 565 \approx 2{,}4$ против $\tilde{S} = 3210 / 1477 \approx 2{,}2$), зато при $\varkappa = 0{,}2$ происходит его резкое увеличение ($\tilde{S} = 11860 / 1496 \approx 7{,}9$). Максимум величины S_{max} изменяется аналогичным образом (23, 22 и 61 соответственно).

6. Среднее значение $\tilde{J}$ числа итераций LH-метода на одну стартовую точку принимает значения соответственно $\tilde{J} = 66723 / 1364 \approx 494$; $1744759 / 3210 \approx 544$; $9487617 / 11860 \approx 800$. Среднее значение числа итераций LH-метода на одну решаемую задачу составляет 118, 1181 и 6450 соответственно. Это означает, что игры решать становится труднее, общее время решения тестируемого семейства 1500 игр для $\varkappa = 0$; 0,1; 0,2 составило соответственно 2; 48; 278 минут. По-видимому, следовало для двух последних строк табл. 1 значительно уменьшить значение параметра I, например до 1.

В табл. 2 приведены результаты тестирования семейства 1500 игр при помощи гибридного метода для трех значений параметра взаимозависимости ($\varkappa \in \{0; 0{,}1; 0{,}2\}$). В табл. 2 приняты обозначения: $1 \leqslant p \leqslant 15$ – номер серии (обычно он опускается); m, n – размеры биматричной игры; множество $R = R^{(p)} = \{1,\ldots,\overline{R}\}$ разбито на пять непересекающихся (возможно, пустых) подмножеств: $R = R_{PS} \cup R_{2LP} \cup R_{LH1} \cup R_{LH2} \cup R_{NO}$, где $k_{PS} = | R_{PS} |$ – число игр, у которых имеются решения в чистых стратегиях, такие игры не нужно решать гибридным методом, $k_{2LP} = | R_{2LP} |$, $k_{LH1} = | R_{LH1} |, k_{LH2} = | R_{LH2} |$ – число игр, решенных соответственно 2LP -, LH1- или LH2-методом; $k_{NO} = | R_{NO} |$ – число игр, которые не удалось решить гибридным методом; итого, $k_{PS} + k_{2LP} + k_{LH1} + k_{LH2} + k_{NO} = \overline{R} = 100$ игр; $\rho = R \setminus R_{PS}$ – множество игр, решаемых гибридным методом. Для $\pi \in \rho$ величины s_π, j_π и t_π – число стартовых точек, число итераций и время, соответственно, понадобившиеся гибридному методу для нахождения решений игр серии p и/или установления факта, что решения серии p игр не найдены, при этом для $\pi \in R_{NO}$ значения $s_\pi = m$, $j_\pi \sim \overline{I}(m+n)m$. Величины S_{sum} – общее число стартовых точек, S_{Hyb} – общее число перспективных стартовых точек, J_{2LP} – число итераций (пар задач линейного программирования), совершенных 2LP-методом, $J_{LH} = J_{LH1} + J_{LH2}$ – суммарное число итераций симплексного типа в алгоритмах LH1 и LH2, *Time* – время (в секундах), потребовавшихся гибридному методу для поиска решения ρ игр, при этом выполняются формулы (12) и

$$J_{LH} = \sum_{\pi \in R_{LH1}} j_\pi + \sum_{\pi \in R_{LH2}} j_\pi.$$

Анализ данных, представленных в табл. 2, дает возможность сделать следующие выводы.

1. Как и в случае LH-метода, гибридному методу при $\varkappa = 0$ решать пришлось лишь 565 игр, при $\varkappa = 0{,}1$ – 1477 игр, при $\varkappa = 0{,}2$ – 1496 игр.

2. При $\varkappa \leqslant 0{,}1$ гибридным методом были решены все игры семейства, при $\varkappa = 0{,}2$ не удалось решить 60 игр из 1500, что в 2,6 раза больше, чем при решении LH-методом. Тем не менее гибридному методу удалось решить несколько игр (см. две последние строки табл. 2), с которыми LH-метод не справился.

3. В каждой серии игр представлены все три типа решений: при $\varkappa = 0$ получено $k_{2LP} = 187$, $k_{LH1} = 313$, $k_{LH2} = 65$; при $\varkappa = 0{,}1$ имеем $k_{2LP} = 429$, $k_{LH1} = 760$, $k_{LH2} = 288$; при $\varkappa = 0{,}2$ получено $k_{2LP} = 387$, $k_{LH1} = 717$, $k_{LH2} = 332$.

4. При увеличении параметра $\varkappa$ в рассматриваемом диапазоне происходит рост среднего числа $\tilde{S}$ стартовых точек на одну решаемую задачу. Так, при $\varkappa = 0$ значение $\tilde{S} = 1282 / 565 \approx 2{,}3$; при $\varkappa = 0{,}1$ имеем $\tilde{S} = 7440 / 1477 \approx 5$, при $\varkappa = 0{,}2$ значение $\tilde{S} = 20960 / 1496 \approx 14$. Падает доля S_{Hyb} / S_{sum} перспективных стартовых точек по отношению к общему числу стартовых точек (0,9; 0,8; 0,5).

Таблица 2. Зависимость от параметра $\varkappa$ результатов решения гибридным методом тестового набора 100 игр

$\varkappa$	№ п/п	Размеры		Результат					Общее число				*Time*, с
		m	n	k_{PS}	k_{2LP}	k_{LH1}	k_{LH2}	k_{NO}	S_{sum}	S_{Hyb}	J_{2LP}	J_{LH}	
0	1	20	20	63	18	16	3	0	69	53	222	4607	46,88
	2	20	40	66	13	15	6	0	65	58	243	6912	52,70
	3	20	60	66	12	18	4	0	54	50	189	5645	44,88
	4	20	80	60	9	25	6	0	77	67	281	9085	68,56
	5	20	100	71	9	18	2	0	65	60	256	9492	65,60
	6	40	40	61	12	22	5	0	86	73	310	10648	71,53
	7	40	60	57	10	29	4	0	103	87	361	13704	87,67
	8	40	80	66	10	22	2	0	71	58	260	9880	64,32
	9	40	100	63	19	15	3	0	78	69	293	12014	74,87
	10	60	60	67	9	18	6	0	86	80	293	12937	78,06
	11	60	80	55	16	24	5	0	107	98	381	15788	98,36
	12	60	100	59	10	28	3	0	120	112	446	21114	121,65
	13	80	80	65	14	17	4	0	82	80	302	15192	80,19
	14	80	100	60	15	21	4	0	101	95	366	18587	101,11
	15	100	100	56	11	25	8	0	118	114	431	22045	119,06
	Сумма			935	187	313	65	0	1282	1154	4665	187650	1175,44
0,1	1	20	20	12	33	41	14	0	171	132	585	13973	119,96
	2	20	40	3	28	45	24	0	241	205	939	30474	196,98
	3	20	60	6	30	45	19	0	301	241	1203	45883	268,53
	4	20	80	0	24	57	19	0	325	261	1316	53932	309,43
	5	20	100	0	24	57	19	0	406	333	1700	73112	408,45
	6	40	40	0	36	43	21	0	337	260	1355	54860	299,66
	7	40	60	0	31	49	20	0	424	339	1794	83235	409,25
	8	40	80	0	30	52	18	0	436	347	1901	97097	459,98
	9	40	100	2	33	49	16	0	557	435	2588	146162	646,64
	10	60	60	0	25	51	24	0	508	406	2298	120700	528,63
	11	60	80	0	23	62	15	0	556	429	2621	152741	637,68
	12	60	100	0	26	53	21	0	741	552	3562	224043	982,78
	13	80	80	0	30	56	14	0	787	599	3752	238982	952,52
	14	80	100	0	31	51	18	0	711	504	3453	236355	876,46
	15	100	100	0	25	49	26	0	939	644	4727	338795	1196,32
	Сумма			23	429	760	288	0	7440	5687	33794	1910344	8923,27
0,2	1	20	20	4	21	49	26	0	264	177	1014	28583	204,07
	2	20	40	0	31	45	24	0	318	248	1366	50608	280,67
	3	20	60	0	26	53	19	2	392	298	1788	76654	396,85
	4	20	80	0	21	56	21	2	471	349	2243	105645	518,62
	5	20	100	0	31	44	24	1	605	433	2894	141391	676,81
	6	40	40	0	34	44	22	0	629	436	3040	149248	658,27
	7	40	60	0	21	50	27	2	901	590	4541	258694	1017,54
	8	40	80	0	21	44	26	9	1255	784	6676	424411	1560,31
	9	40	100	0	19	51	23	7	1380	780	7523	515637	1791,57
	10	60	60	0	26	47	26	1	1723	988	9288	624208	2114,92
	11	60	80	0	21	51	24	4	1487	801	8428	627664	2023,27
	12	60	100	0	31	44	17	8	2353	1100	13980	1142260	3309,40
	13	80	80	0	30	49	14	7	2327	1059	13842	1157629	3235,16
	14	80	100	0	21	48	27	4	2897	1131	18014	1638420	4193,62
	15	100	100	0	33	42	12	13	3988	1451	25706	2563992	6166,30
	Сумма			4	387	717	332	60	20960	10625	120343	9504944	28147,38

5. Введем следующие обозначения: $\tilde{J}_{2LP}$ и $\bar{J}_{2LP}$ – среднее значение числа итераций 2LP-метода (пар задач линейного программирования) на одну стартовую точку и на одну решаемую задачу; $\tilde{J}_{LH}$ и $\bar{J}_{LH}$ – среднее значение числа итераций LH-метода на одну перспективную стартовую точку и на одну решаемую задачу соответственно. Тогда

$\varkappa = 0:$

$$\tilde{J}_{2LP} = \frac{4665}{1282} \approx 4, \quad \bar{J}_{2LP} = \frac{4665}{187} \approx 25, \quad \tilde{J}_{LH} = \frac{187050}{1154} \approx 163, \quad \bar{J}_{LH} = \frac{187050}{3313+65} \approx 496,$$

$\varkappa = 0{,}1:$

$$\tilde{J}_{2LP} = \frac{33794}{7440} \approx 5, \quad \bar{J}_{2LP} = \frac{33794}{429} \approx 79, \quad \tilde{J}_{LH} = \frac{19 * 10344}{5687} \approx 336, \quad \bar{J}_{LH} = \frac{1910344}{760+288} \approx 1823,$$

$\varkappa = 0{,}2:$

$$\tilde{J}_{2LP} = \frac{120343}{20960} \approx 6, \quad \bar{J}_{2LP} = \frac{120343}{387} \approx 311, \quad \tilde{J}_{LH} = \frac{9504944}{10625} \approx 895, \quad \bar{J}_{LH} = \frac{9504944}{717+332+60} \approx 8571.$$

Общее время решения гибридным методом тестируемого семейства 1500 игр для $\varkappa = 0; 0{,}1; 0{,}2$ составило соответственно 20; 138; 469 минут.

Сравним результаты решения игр с независимыми матрицами (при $\varkappa = 0$), полученные при помощи LH- и гибридного метода. Все 1500 тестовых задач были решены. Рассмотрим подробно серию 100×100 игр. В 56 случаях игры имели точку Нэша в чистых стратегиях. В 11 случаях удалось получить решение уже после применения 2LP-метода, для 25 игр пришлось дополнительно использовать LH1-алгоритм, остальные 8 игр были решены после добавления дополнительного столбца. Был сделан 431 шаг 2LP-метода, примерно 10 шагов (431/(11+25+8)) в среднем на одну решаемую игру (без учета игр, имеющих точку Нэша в чистых стратегиях). Всего было использовано 118 стартовых точек, из них перспективными оказалось 114 стартовых точек. Среднее количество начальных точек на одну игру равно 3, максимальное – 10. Для решения 33 игр было сделано 22045 итераций симплексного типа.

В то же время LH-методу для нахождения решения 44 игр пришлось просмотреть 142 стартовые точки, сделав 11242 итерации LH-процедуры.

Мы отмечали ранее, что результат решения зависит от значения параметра I. Имеется зависимость и от других параметров. Так, при $\varkappa = 0$, $\bar{F} = 10^{-5}$ и $\bar{\Delta} = 1$, перебрав 313 стартовых точек, 2LP-метод решил все 44 игры серии 100×100 без помощи LH-процедуры. Следовательно, игры с независимыми матрицами можно решать 2LP-методом, не привлекая LH-процедуру и построение дополнительного столбца, но затратив значительно больше итераций и времени (в упомянутом примере число использованных стартовых точек возросло более чем в три раза, а время – на одну четверть).

Как уже отмечалось ранее, имеются игры, решая которые, гибридный метод выигрывает у LH-метода. Покажем, что гибридный метод способен оказать конкуренцию LH-методу при решении игр большого размера. Проиллюстрируем это утверждение.

В табл. 3 приведены полученные при помощи LH-метода и гибридного метода результаты решения нескольких биматричных игр большого размера. В ней приняты следующие обозначения: *LH* – сокращение от метода Лемке – Хаусона, *Hyb* – сокращение от гибридного метода; параметр I задает предельно допустимое число (равное $I(m+n)$) итераций в LH-методе; m, n – размеры биматричной игры; $\bar{R}$ – число игр в серии, $k_{PS} = | R_{PS} |$ – число игр, у которых имеются решения в чистых стратегиях, множество $\rho = R \setminus R_{PS}$, k_{NO} – число нерешенных задач в серии игр.

Как и в табл. 1, 2, s_π, j_π и t_π – число стартовых точек, число итераций и время, соответственно, понадобившиеся LH-методу или гибридному методу для нахождения решения игр серии p и/или установления факта, что решения игр серии p не найдены, величины S_{sum} – общее число стартовых точек, J_{sum} – суммарное число итераций симплексного типа, *Time* – суммарное время (в секундах) вычисляются по формулам (12).

Таблица 3. Сравнение результатов, полученных при помощи LH-метода и гибридного метода, при решении биматричных игр большого размера

№ п/п	Размеры		ρ	ϰ	Метод		Результат			*Time*, с
	m	*n*			*LH* или *Hyb*	*I*	k_{NO}	S_{sum}	J_{sum}	
1	400	800	1	0	*LH*	500	0	5	2400221	6051
	400	800	1	0	*LH*	0,25	0	5	1421	19
	400	800	1	0	*Hyb*	–	0	3	1468	17
2	400	800	1	0,1	*LH*	50	**1**	400	480000	4831
	400	800	1	0,1	*Hyb*	–	0	208	381653	1382
3	500	500	1	0	*LH*	10	0	2	5707	70
	500	500	1	0	*Hyb*	–	0	9	3453	17
4	500	500	1	0,1	*LH*	10	**1**	500	500000	45727
	500	500	1	0,1	*Hyb*	–	0	39	74978	179
5	500	500	4	0	*LH*	50	0	4	16423	380
	500	500	4	0	*Hyb*	–	0	2726	17448	99
6	1000	1000	3	0	*LH*	5	0	27	260235	517670
	1000	1000	3	0	*LH*	0,25	0	27	12235	18810
	1000	1000	3	0	*Hyb*	–	0	7	6761	6020
7	2000	2000	36	0	*LH*	0,1	0	4761	187833	6711
	2000	2000	36	0	*Hyb*	–	0	221	266452	5982
8	3000	3000	39	0	*LH*	0,01	0	6004	358826	19374
	3000	3000	39	0	*Hyb*	–	0	328	485359	17238
9	5000	5000	5	0	*LH*	0,01	0	880	87732	13566
	5000	5000	5	0	*Hyb*	–	0	66	118317	27499

Проанализировав данные в табл. 3, приходим к следующим выводам.

1. Продемонстрировано еще раз, как удачный выбор параметра I влияет на скорость работы LH-метода. Так, для игры 400×800 при значении параметра $I = 500$, следовательно, при максимуме итераций $I(m+n) = 600000$ число итераций составило $4 \times 600000 + 221 = 2400221$, а при $I = 0{,}25$ число совершенных итераций равно $4 \times 0{,}25(400+800) + 221 = 1421$. В обоих случаях результат был достигнут на пятой стартовой точке после выполнения 221 итерации при колоссальной разнице во времени. В то же время гибридный метод использовал только три стартовые точки и получил решение игры на шаге LH1, завершив 1464 итерации за 17 секунд.

2. Среди игр, приведенных в табл. 3, две игры, 400×800 и 500×500, при $\varkappa = 0{,}1$ не удалось решить при различных значениях параметра I, используя LH-метод. В то же время эти же игры были успешно решены гибридным методом: игра 400×800 после обработки 208 стартовых точек за 381653 итерации и 1382 секунды; игра 500×500 – после обработки 39 стартовых точек за 74978 итераций и 179 секунд.

3. Оба алгоритма решают успешно игры с независимыми матрицами с примерно одинаковой эффективностью при удачном подборе параметра I (см., например, строки табл. 3 с номерами 7–9). Так, при решении задачи 3000×3000 гибридным методом было использовано 328 стартовых точек, выполнено 485359 итераций за 17238 секунд. А LH-методу пришлось использовать 6004 стартовых точек, выполнить 358826 итераций за 19374 секунды при параметре $I = 0{,}01$ (значение $I(m+n) = 60$).

Из выше сказанного можно сделать заключение о том, что гибридный метод способен оказать конкуренцию 2LP-методу и LH-методу, а в ряде случаев находит решение игры, когда 2LP-методу и LH-методу не удается это сделать.

СПИСОК ЛИТЕРАТУРЫ

Гольштейн Е.Г., Малков У.Х., Соколов Н.А. (2013). Об одном численном методе решения биматричных игр // *Экономика и математические методы*. Т. 49. № 4. С. 94–104.

IBM ILOG CPLEX Optimization Studio (2011). [Электронный ресурс] Режим доступа: http://www-03.ibm.com/software/products /ru/ibmilogcpleoptistud, свободный. Загл. с экрана. Яз. англ. (дата обращения: июль 2017 г.).

Lemke C.E., Howson J.T. Jr. (1964). Equilibrium Points of Bimatrix Games // *Journal of the Society for Industrial and Applied Mathematics*. Vol. 12. P. 778–780.

MATLAB (2012). The Language of Technical Computing. [Электронный ресурс] Режим доступа: http://www.mathworks.com/products /matlab/, свободный. Загл. с экрана. Яз. англ. (дата обращения: июль 2017 г.).

Поступила в редакцию
24.05.2017 г.

REFERENCES (with English translation or transliteration)

Golshtein E.G., Malkov U.H., Sokolov N.A. (2013). A Numerical Method for Solving Bimatrix Games. *Economics and Mathematical Methods*, 49, 4, 94–104.

IBM ILOG CPLEX Optimization Studio (2011). Available at: http://www-03.ibm.com/software/products /ru/ibmilogcpleoptistud, свободный. Загл. с экрана. Яз. англ. (дата обращения: июль 2017 г.).

Lemke C.E., Howson J.T. Jr. (1964). Equilibrium Points of Bimatrix Games. *Journal of the Society for Industrial and Applied Mathematics*, 12, 778–780.

MATLAB (2012). The Language of Technical Computing. Available at: http://www.mathworks.com/products / matlab/, свободный. Загл. с экрана. Яз. англ. (дата обращения: июль 2017 г.).

Received 24.05.2017

HYBRID METHOD FOR SOLVING BI–MATRIX GAMES

E.G. Golshtein[i], U.H. Malkov[ii], N.A. Sokolov[iii]

Abstract. In order to find optimal mixed strategies of a bi-matrix game one can use the approximate algorithm for solving bi-matrix games (2LP-algorithm) and/or the Lemke–Howson algorithm (LH-algorithm). When solving a bi-matrix game the 2LP-algorithm provides finding the global minimum of the Nash function by processing (numerous) local minima of the same function. The point is that every successive (alternating) minimization of this function with respect to one of the two variables (while the other variable is fixed) can be easily reduced to a linear programming problem. By making use of a trial of the initial pure strategies and solving two linear programs at each iteration the 2LP-algorithm either establishes the game's exact solution (if a complementarity slackness condition holds) or finds an approximation to the set of Nash points (when the complementarity slackness condition is slightly broken). The algorithm is very easy to implement, however, it loses its efficiency whenever the payoff matrices are scarce and/or (mutually) linearly dependent on each other. On the contrary, the LH-algorithm reduces the bi-matrix game to the solution of a related linear equations system. Starting from the basis of the unit vectors, the method makes simplex-type steps aiming to decrease the number of broken complementarity slackness conditions. As a rule (but not always) the algorithms finds the exact solution of the game. The proposed hybrid method uses the LH-algorithm to re-optimize an approximate solution produced by the 2LP-algorithm starting with the basis of the latter. The efficiency of the Lemke–Howson algorithm and the proposed hybrid approach has proved to be about the same. Moreover, the hybrid method has managed to find exact solutions to several test games for which both the 2LP- and LH-procedures failed.

Keywords: bi-matrix games, convex structures, pure strategies, mixed strategies, Nash points, Nash function, complementarity slackness condition, Lemke-Howson algorithm, hybrid method.

Classification JEL: C02, C72.

[i] **Evgeny G. Golshtein** – chief researcher, doctor of physico-mathematical Sciences, prof., Central Economics and Mathematics Institute, Russian Academy of Sciences, Nakhimovskii prospekt, B. 47, Moscow, 117418, Russia; golshtn@cemi.rssi.ru.

[ii] **Ustav H. Malkov** – leading researcher, candidate of physico-mathematical Sciences, Central Economics and Mathematics Institute, Russian Academy of Sciences, Nakhimovskii prospekt, B. 47, Moscow, 117418, Russia; malkov@cemi.rssi.ru.

[iii] **Nikolay A. Sokolov** – leading researcher, candidate of physico-mathematical Sciences, Central Economics and Mathematics Institute, Russian Academy of Sciences, Nakhimovskii prospekt, B. 47, Moscow, 117418, Russia; sokolov@cemi.rssi.ru.