

УДК 519.833.3

О формировании процесса обучения решению биматричных игр с помощью алгоритма Лемке-Хоусона

Д.С. Набатова

Аннотация

В статье изложено описание компьютерного практикума для студентов Московского авиационного института по вычислительному алгоритму Лемке-Хоусона решения биматричной игры. Рассматриваются основные аспекты вычислений и пример с представлением рабочих окон программы.

Ключевые слова

биматричные игры; обучающие программы; компьютерный практикум; вычислительный алгоритм; программное обеспечение; многооконная среда; пример вычислений.

Введение

При подготовке специалистов в области прикладной математики важным аспектом является не только знакомство с теоретическими основами таких дисциплин, как теория управления, системный анализ, теория игр, но и навыки использования вычислительных алгоритмов и умение решать с их помощью конкретные задачи. С этой целью разрабатываются программы компьютерных занятий, на которых студенты могли бы ознакомиться с теоретическими основами алгоритмов, проверить качество знаний по данному предмету и решить с помощью готового программного обеспечения конкретную задачу. При подготовке компьютерного практикума были учтены все требования, предъявляемые к таким работам.

Вычислительный алгоритм Лемке-Хоусона как и сами биматричные игры нечасто рассматриваются в учебных курсах, т.к. обычно теоретические курсы ограничиваются

рассмотрением антагонистических игр. Для остальных игровых моделей вычислительных алгоритмов практически нет, а имеющиеся алгоритмы практически не представлены в стандартном программном обеспечении. К таким алгоритмам можно отнести и алгоритм Лемке-Хоусона. Тем не менее, изучение этого алгоритма позволяет поднять методику решения конечных игр на новый уровень, осуществив переход от традиционной двойственной задачи линейного программирования для игр с одной платежной матрицей, к более сложной процедуре поиска решения для биматричной игры.

Описание алгоритма

Рассмотрим биматричную игру.

$$\Gamma = \{X, Y, A, B, m, n\}$$

$X = \{x_1, x_2, \dots, x_m\}$ - множество чистых стратегий первого игрока

$Y = \{y_1, y_2, \dots, y_n\}$ - множество чистых стратегий второго игрока

(m, n) – система чисел, определяющая размерность платежных матриц и формат игры.

Все значения выигрышей, которые получают игроки 1 и 2 при выборе ими своих стратегий с номерами i и j определяются из двух матриц A и B , размерностью $m \times n$. Элементы матрицы A определяют выигрыш 1 игрока, а матрицы B – второго. Решить игру, значит определить пару (x^*, y^*) , которая удовлетворяла бы определению ситуации равновесия, и значения выигрышей V_A и V_B . В чистых стратегиях ситуация равновесия существует не всегда, поэтому при решении конечных игр рассматривают их смешанные расширения и определяют ситуацию равновесия для них. Смешанным расширением биматричной игры называется игра, где стратегии игроков 1 и 2 определяются двумя векторами:

$$x = \{x_1, x_2, \dots, x_m\} \text{ и } y = \{y_1, y_2, \dots, y_n\}$$

из фундаментальных симплексов X и Y

$$X = \left\{ x = (x_1, x_2, \dots, x_m) \mid x_i \geq 0, 1 \leq i \leq m, \sum_{i=1}^m x_i = 1 \right\}$$

$$Y = \left\{ y = (y_1, y_2, \dots, y_n) \mid y_j \geq 0, 1 \leq j \leq n, \sum_{j=1}^n y_j = 1 \right\}$$

Ситуация равновесия для смешанного расширения определяется неравенствами:

$$\sum_{i=1}^m \sum_{j=1}^n x_i^* \cdot y_j^* \cdot a_{ij} \geq \sum_{i=1}^m \sum_{j=1}^n x_i \cdot y_j^* \cdot a_{ij} = H_1(x, y^*) \text{ для } \forall x \in X$$

$$\sum_{i=1}^m \sum_{j=1}^n x_i^* \cdot y_j^* \cdot b_{ij} \geq \sum_{i=1}^m \sum_{j=1}^n x_i^* \cdot y_j \cdot b_{ij} = H_2(x^*, y) \text{ для } \forall y \in Y$$

Функции выигрыша смешанного расширения $H_1(x, y)$ и $H_2(x, y)$ определены на множестве $X \times Y$. Такие ситуации называют ситуациями равновесия по Нэшу. С содержательной стороны равновесие можно истолковать следующим образом: в случае, если один из игроков не придерживается равновесной стратегии, он получает меньший выигрыш, при условии, что второй участник игры выбирает для себя стратегию равновесия.

Вычислительный алгоритм Лемке-Хоусона позволяет определить элементы векторов, составляющих ситуацию равновесия по Нэшу по элементам платежных матриц А и В.

Можно показать [1], что задача определения ситуации равновесия (x^0, y^0) по Нэшу сводится к определению параметра α и пары (x^*, y^*) - векторов, удовлетворяющих следующим соотношениям:

$$x^* = (x_1^*, x_2^*, \dots, x_m^*), y^* = (y_1^*, y_2^*, \dots, y_n^*), x_i^* \geq 0, 1 \leq i \leq m, y_j^* \geq 0, 1 \leq j \leq n$$

$$a'_{ij} = \alpha - a_{ij} > 0, 1 \leq i \leq m, 1 \leq j \leq n, b'_{ij} = \alpha - b_{ij} > 0, 1 \leq i \leq m, 1 \leq j \leq n,$$

$$\sum_{j=1}^n a'_{ij} \cdot y_j^* \geq 1, 1 \leq i \leq m, \quad \sum_{i=1}^m b'_{ij} \cdot x_i^* \geq 1, 1 \leq j \leq n,$$

$$\sum_{i=1}^m x_i^* \cdot \left(\sum_{j=1}^n a'_{ij} \cdot y_j^* - 1 \right) = 0,$$

$$\sum_{j=1}^n y_j^* \cdot \left(\sum_{i=1}^m b'_{ij} \cdot x_i^* - 1 \right) = 0,$$

$$x_i^0 = \frac{x_i^*}{\sum_{i=1}^m x_i^*}, 1 \leq i \leq m; y_j^0 = \frac{y_j^*}{\sum_{j=1}^n y_j^*}, 1 \leq j \leq n.$$

Идея алгоритма Лемке-Хоусона состоит в направленном переборе пар векторов (x, y) , определяемых $n+m-1$ равенствами до достижения точки, в которой выполняются $(m+n)$ условий.

Итерационная процедура симплекс-преобразования идет отдельно по множеству X и множеству Y . Составляются таблицы: из столбцов матрицы A и векторов базиса R^n - таблица A^* , из строк матрицы B и векторов базиса R^m - таблица B^* .

Таблица A .

	a_1	a_2	.	.	.	a_m	e_1	e_2	.	.	.	e_n
e_1	a_{11}	a_{21}	.	.	.	a_{m1}	1	0	.	.	.	0
e_2	a_{12}	a_{22}	.	.	.	a_{m1}	0	1	.	.	.	0
.
e_n	a_{1n}	a_{2n}	.	.	.	a_{mn}	0	0	.	.	.	1

Индексы e_1, e_2, \dots, e_n в левом наружном столбце указывают на то, что все векторы $a_1, \dots, a_m, e_1, \dots, e_n$ выражены именно через этот базис (как и в таблицах симплекс-метода).

Таблица B .

	b_1	b_2	.	.	.	b_n	f_1	f_2	.	.	.	f_m
f_1	b_{11}	b_{12}	.	.	.	b_{1n}	1	0	.	.	.	0
f_2	b_{21}	b_{22}	.	.	.	b_{2n}	0	1	.	.	.	0
.
f_m	b_{m1}	b_{m2}	.	.	.	b_{mn}	0	0	.	.	.	1

Определяется $(x^{(0)}, y^{(0)})$ - начальная точка. По отличным от нуля элементам начальной точки определяются те векторы a_i и b_j , которые будут введены в базис, вычисляются новые симплекс-таблицы A^* и B^* . Каждый шаг алгоритма определяет множество векторов:

$$x_i^{(k)} = x_i^{(k-1)} + \lambda_i p^i, \quad 1 \leq i \leq m,$$

$$y_j^{(k)} = y_j^{(k-1)} + \mu_j q^j, \quad 1 \leq j \leq n$$

где λ и μ вычисляются следующим образом:

вычислим $\xi_i = \sum_{j=1}^n a_{ij} \cdot y_j^{(0)}$, $i = \overline{1, m}$,

из всех отрицательных элементов таблицы A^* $a_{kj} < 0$, $q^{jr} < 0$ найдем

$$\min \left\{ -\frac{(\xi_k - 1)}{a_{kj}}; -\frac{y_r^{(0)}}{q^{jr}} \right\} = \lambda^*$$

из всех положительных элементов $a_{sj} > 0$, $q^{jt} > 0$ найдем

$$\max \left\{ -\frac{(\xi_k - 1)}{a_{sj}}; -\frac{y_t^{(0)}}{q^{jt}} \right\} = \lambda^{**}$$

В столбец с λ вводится тот элемент из $(\lambda^*, \lambda^{**})$, который отличен от 0. Если они оба равны 0, то вводится 0. Из таблицы A^* можно определить n новых точек, принадлежащих множеству Y .

Аналогично для таблицы B^* : $\eta_j = \sum_{i=1}^m b_{ij} \cdot x_i^{(0)}$, $j = \overline{1, n}$, из всех отрицательных

элементов таблицы B^* $b_{is} < 0$, $p^{ir} < 0$ найдем

$$\min \left\{ -\frac{(\eta_k - 1)}{b_{ik}}; -\frac{x_r^{(0)}}{p^{ir}} \right\} = \mu^*$$

из всех положительных элементов $b_{is} > 0$, $p^{it} > 0$ найдем

$$\max \left\{ -\frac{(\eta_s - 1)}{\beta_{is}}; -\frac{x_t^{(0)}}{p^{it}} \right\} = \mu^{**}$$

Для того чтобы вычисления были более упорядоченными, имеет смысл выбирать $y^{(0)}$ по первому столбцу матрицы A , т.е. для $j=1$. Однако если существует дополнительная информация о номерах компонент, отличных от 0 в ситуации равновесия, то при выборе номера j имеет смысл ее учитывать. Для произвольного случая положим $j=1$ и определим

$y^{(0)} = (1/a_{i1}, 0, 0, \dots, 0)$. Пусть например $i = 3$. Очевидно, что для $y^{(0)}$ выполняется n соотношений из определения множества Y :

$$y_j^{(0)} = 0, \quad j = 2, 3, 4, \dots, n, \quad \sum_{j=1}^n a_{3j} \cdot y_j^{(0)} = \frac{a_{31}}{a_{31}} = 1$$

Определим $x^{(0)}$ по строке матрицы B с номером $i = 3$ для нашего случая $x^{(0)} = (0, 0, 1/k, 0, \dots, 0)$. Это будет $k = \min(b_{31}, b_{32}, \dots, b_{3n})$. Пусть $k = b_{32}$. Тогда:

$$x_i^{(0)} = 0, \quad i = 1, 2, 4, \dots, m, \quad \sum_{i=1}^m b_{i2} \cdot x_i^{(0)} = \frac{b_{32}}{b_{32}} = 1.$$

Для того чтобы пара $(x^{(0)}, y^{(0)})$ была ситуацией равновесия, необходимо, чтобы для нее выполнялись условия:

$$\sum_{j=1}^n y_j^{(0)} \cdot (\sum_{i=1}^m b_{ij} \cdot x_i^{(0)} - 1) = 0$$

$$\sum_{i=1}^m x_i^{(0)} \cdot (\sum_{j=1}^n a_{ij} \cdot y_j^{(0)} - 1) = 0$$

Это произошло бы в том случае, если бы номер минимального элемента, определяющего $x^{(0)}$ (в нашем примере это $j = 2$) совпадал бы с номером j -той строки таблицы A , по которой выбрана начальная точка ($j = 1$). Если бы это условие было выполнено, то нормируя $(x^{(0)}, y^{(0)})$ можно было бы получить ситуацию равновесия по Нэшу. В данном случае это не так, поэтому перейдем к следующему шагу.

Найдем множества, ранее обозначенные через $p(x^{(0)})$ и $q(y^{(0)})$:

$$p(x^{(0)}) = \left\{ f_i, b_j \mid x_i = 0, \sum_{i=1}^m x_i \cdot b_{ij} - 1 = 0 \right\}$$

$$q(y^{(0)}) = \left\{ e_j, a_i \mid y_j = 0, \sum_{j=1}^n y_j \cdot a_{ij} - 1 = 0 \right\}$$

В данном случае:

$$p(x^{(0)}) = (f_1, f_2, b_2, f_4, \dots, f_m)$$

$$q(y^{(0)}) = (a_3, e_2, \dots, e_n)$$

Заменяем базисы в таблицах А и В на $p(x^0)$ и $q(y^0)$ и пересчитаем новые таблицы A^* и B^* .

Чтобы получить таблицу A^* нужно из базиса исключить вектор e_1 и ввести на его место a_3 . Следовательно, таблица А должна быть преобразована таким образом: вместо a_3 должен появиться столбец, содержащий все, кроме первого, нулевые элементы, а в правой части таблицы вместо e_1 появиться произвольного вида столбец, с отличными от 0 элементами.

Таблица A^*

	a_1	a_2	a_3	a_4	\dots	a_m	e_1	e_2	\dots	e_n	λ
a_3	α_{11}	α_{21}	1	α_{41}	\dots	α_{m1}	q^{11}	0	\dots	0	λ_1
e_2	α_{12}	α_{22}	0	α_{42}	\dots	α_{m2}	q^{21}	1	\dots	0	λ_2
\cdot	\cdot	\cdot	\cdot	\cdot	\dots	\cdot	\cdot	\cdot	\dots	\cdot	\cdot
e_n	α_{1n}	α_{2n}	0	α_{4n}	\dots	α_{mn}	q^{n1}	0	\dots	1	λ_n
	$\xi_1 - 1$	$\xi_2 - 1$	$\xi_3 - 1$	$\xi_4 - 1$	\dots	$\xi_m - 1$	$y_1^{(0)}$	$y_2^{(0)}$	\dots	$y_n^{(0)}$	

Аналогично, чтобы получить таблицу B^* нужно из базиса исключить вектор f_3 и ввести на его место b_2 .

Таблица B^*

	b_1	b_2	\dots	b_n	f_1	f_2	f_3	\dots	f_m	μ
f_1	β_{11}	0	\dots	β_{1n}	1	0	p^{13}	\dots	0	μ_1
f_2	β_{21}	0	\dots	β_{2n}	0	1	p^{23}	\dots	0	μ_2
b_2	β_{31}	1	\dots	β_{3n}	0	0	p^{33}	\dots	0	μ_3
\cdot	\cdot	\cdot	\dots	\cdot	\cdot	\cdot	\cdot	\dots	\cdot	\cdot
f_m	β_{m1}	0	\dots	β_{mn}	0	0	p^{m3}	\dots	1	μ_m
			\dots					\dots		

	$\eta_1 - 1$	$\eta_2 - 1$...	$\eta_n - 1$	$x_1^{(0)}$	$x_2^{(0)}$	$x_3^{(0)}$...	$x_m^{(0)}$	
--	--------------	--------------	-----	--------------	-------------	-------------	-------------	-----	-------------	--

В отличие от классического симплекс-метода выбор начальной точки, следующей точки может быть осуществлен несколькими способами. От того, насколько это удачно сделано, зависит сходимость алгоритма. Кроме этого в некоторых случаях алгоритм может возвращаться в исходную точку. Подобные моменты вычислений требуют контроля извне. Многооконная среда позволяет одновременно реализовывать все аспекты вычислительного алгоритма, осуществлять возврат к уже пройденным этапам вычислений, принимать участие в процессе принятия решений в тех случаях, когда это необходимо. Участие студентов в процессе вычислений позволяет познакомиться более подробно с основными аспектами алгоритма, но избавляет от необходимости производить рутинные арифметические вычисления в большом объеме.

Предлагаемый компьютерный практикум “Вычислительный алгоритм Лемке-Хоусона решения биматричной игры” предназначен для практических занятий по курсам “Теория игр и исследование операций” и “Оптимизация дискретных моделей в механике и макроэкономике”, читаемых на факультете «Прикладная математика и физика» Московского авиационного института.

Практикум состоит из нескольких частей:

- краткое изложение основ алгоритма
- решение численного примера с исходными данными, подготовленными самим студентом, в интерактивном режиме, который позволяет продемонстрировать практические навыки владения алгоритмом и участвовать в процессе принятия решений по тем аспектам вычислений, которые требуют вмешательства извне. Компьютерный практикум выполнен в операционной системе Windows [2].

Описание программы

Перед началом работы необходимо создать файл с исходными данными. Это размеры матриц m и n , а также сами матрицы A и B . После ввода исходных данных появляется рабочее окно программы (рис.1), в котором отражены основные этапы решения задачи, такие как:

- определение параметра α ;
- выбор начальной точки;
- преобразования таблиц A и B ;

- симплекс таблицы A и B;
- выбор следующей точки;
- промежуточные результаты;
- ситуация равновесия;

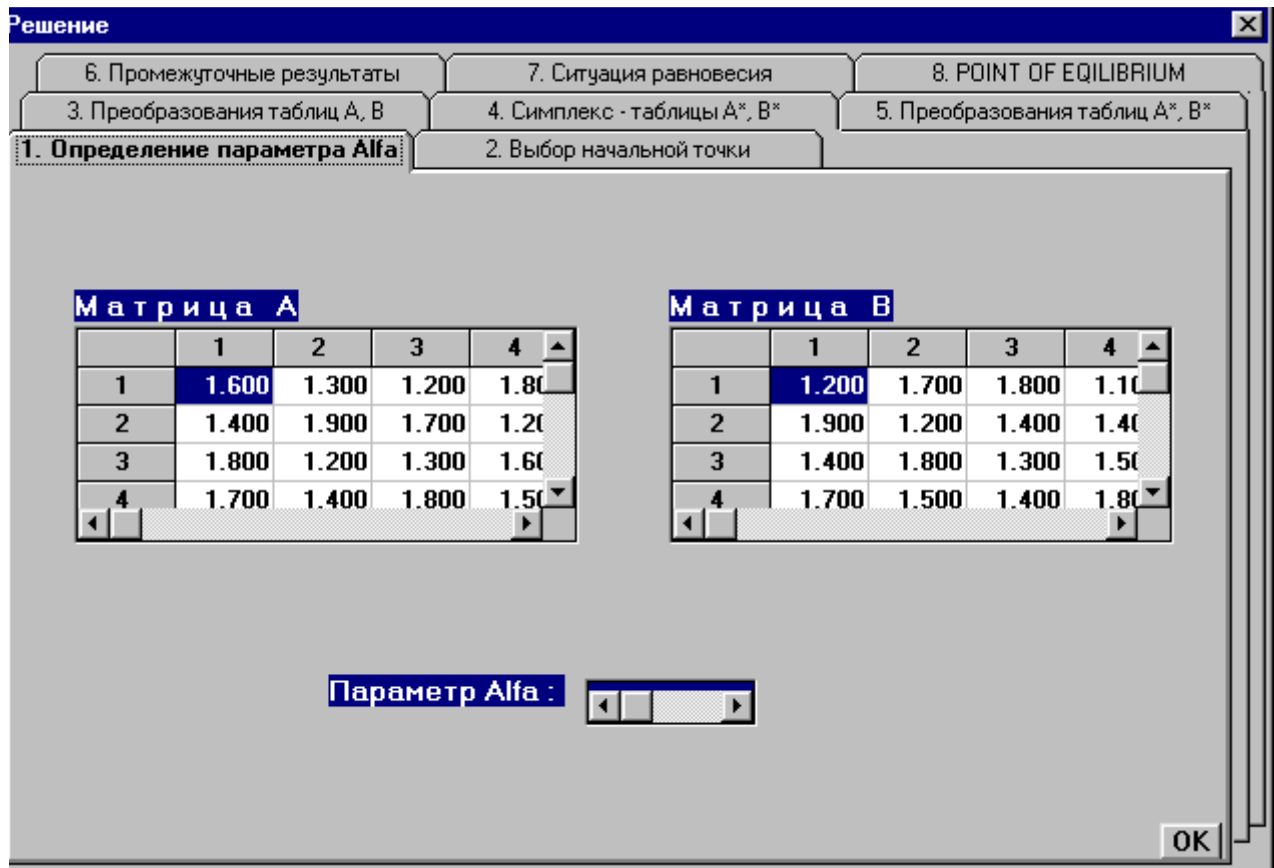


Рис. 1.

При решении конкретного численного примера подразумевается участие пользователя в определенных этапах вычислений, таких как: выбор параметра, выбор строки для вычисления начального приближения, выбор следующей точки из двух возможных, принятие решения о продолжении или завершении расчетов. Все эти действия легко осуществляются с помощью клавиатуры или щелчка мыши в соответствующем месте рабочего окна.

Основные этапы работы рассмотрим на примере.

Определим исходные данные

Размерность матриц 5×4

матрица А

1.600	1.300	1.200	1.800	1.100
1.400	1.900	1.700	1.200	1.300
1.800	1.200	1.300	1.600	1.400
1.700	1.400	1.800	1.500	1.200

Матрица В

1.200	1.700	1.800	1.100	1.500
1.900	1.200	1.400	1.400	1.700
1.400	1.800	1.300	1.500	1.200
1.700	1.500	1.400	1.800	1.000

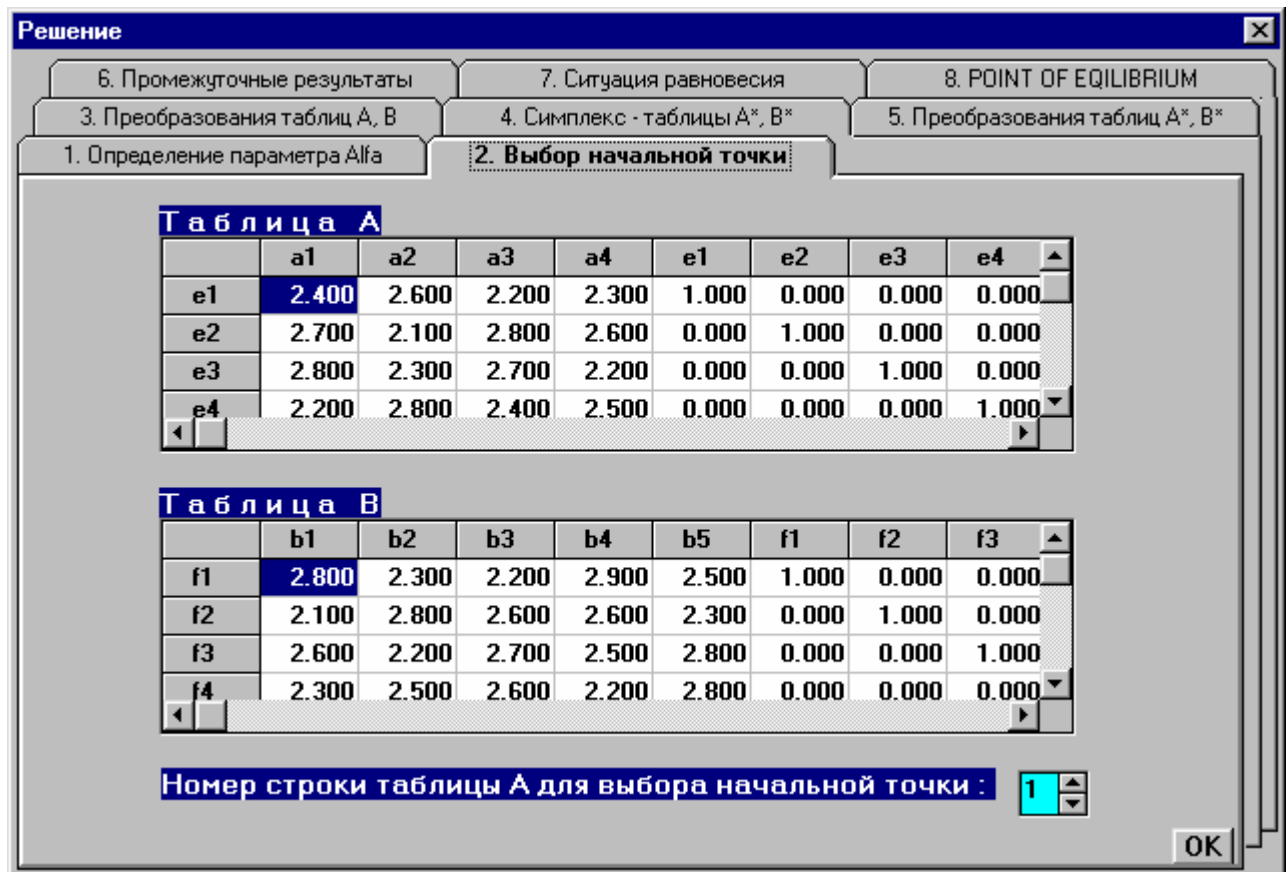


Рис 2.

Начальная точка выбирается по первой строке матрицы А. Для того чтобы можно было ознакомиться со всеми компонентами таблиц, соответствующие рабочие окна снабжены полосами прокрутки, что дает возможность решать задачи достаточно большой размерности.

В последних столбцах таблиц указывается номер столбца матрицы, из которого получено значение λ или μ и признак того, из какой части таблицы «R» - правой, «L» - левой.

Далее появляется сообщение о том, по какой строке матрицы A выбирается начальная точка и какие строки и столбцы исходных таблиц вводятся в базис. На рис.3 представлено окно с базисами начальных точек.

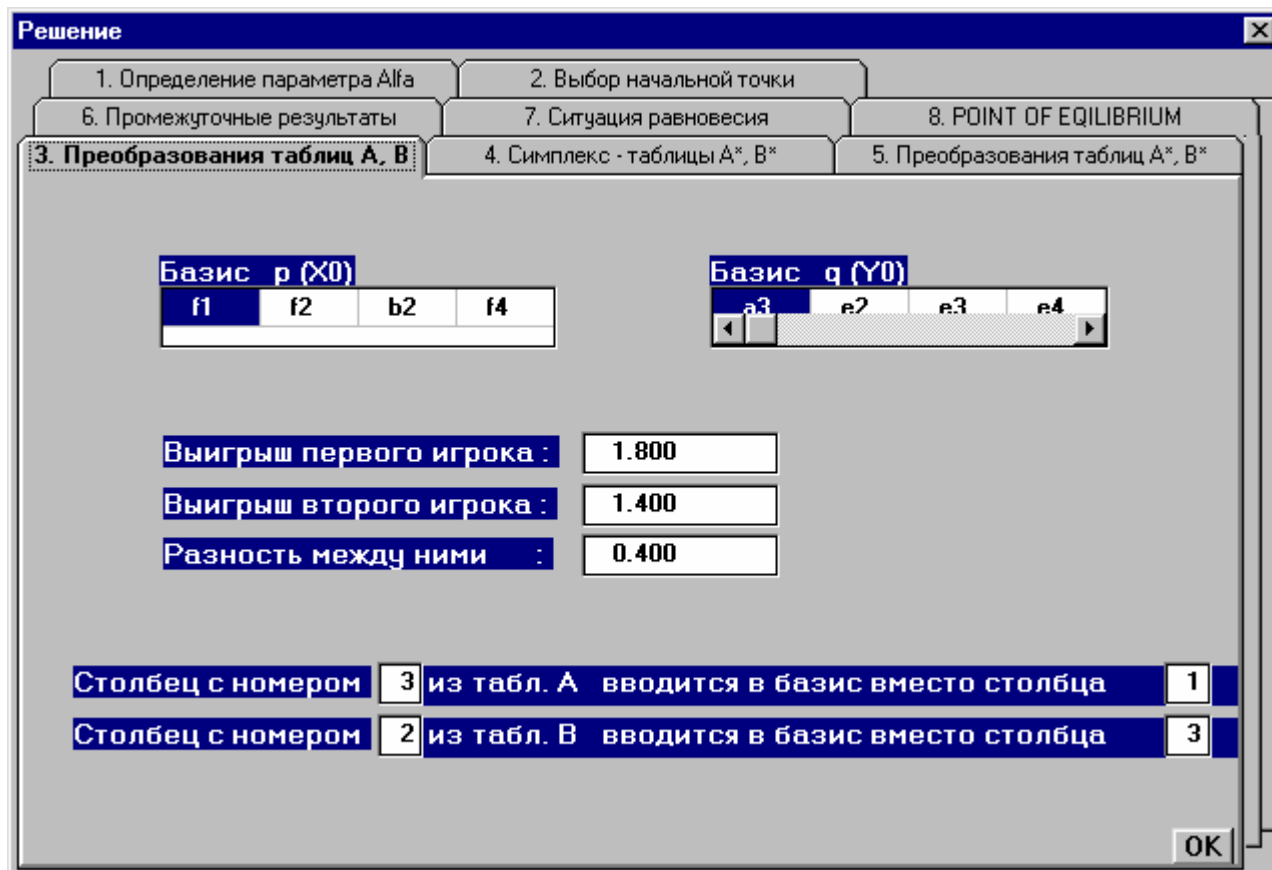


Рис. 3

Затем пересчитываются таблицы в соответствии с указанными преобразованиями. Такая организация вычислений позволяет одновременно отслеживать как процесс преобразований таблиц с матрицами A и B , так и работу с базисами, что в алгоритме является определяющим при получении решения.

Указанные столбцы матриц вводятся в базис. Таблицы A и B пересчитываются в автоматическом режиме. Соответствующее рабочее окно приведено на рис. 4. После расчета симплекс-таблиц вычисляются следующие точки. Их должно быть две. Выбор одной из них осуществляется пользователем с помощью щелчка мыши в соответствующей строке. Иллюстрация этих действий на рис. 5. После выбора преобразовываются симплекс таблицы, опять определяется следующая точка, и вычисления идут до тех пор, пока не будет вычислена ситуация равновесия или алгоритм не возвратится в исходную точку. Вывод об этом можно сделать, анализируя базисы, если базис для текущей точки совпадает с

начальной, значит, начальная точка выбрана неудачно, и нужно проводить вычисления с другим выбором начальной точки.

Решение

1. Определение параметра Alfa 2. Выбор начальной точки

6. Промежуточные результаты 7. Ситуация равновесия 8. POINT OF EQUILIBRIUM

3. Преобразования таблиц A, B **4. Симплекс - таблицы A*, B*** 5. Преобразования таблиц A*, B*

Симплекс - таблица A*

	a1	a2	a3	a4	a3	e2	e3	e4	e5
a3	1.091	1.182	1.000	1.045	0.455	0.000	0.000	0.000	0.000
e2	-0.355	-1.209	0.000	-0.327	-1.273	1.000	0.000	0.000	0.000
e3	-0.145	-0.891	0.000	-0.623	-1.227	0.000	1.000	0.000	0.000
e4	-0.418	-0.036	0.000	-0.009	-1.091	0.000	0.000	1.000	0.000
e5	0.064	-0.373	0.000	0.082	-1.182	0.000	0.000	0.000	1.000

Симплекс - таблица B*

	b1	b2	b3	b4	b5	f1	f2	b2	f4
f1	0.082	0.000	-0.623	0.286	-0.427	1.000	0.000	-1.045	0.000
f2	-1.209	0.000	-0.836	-0.582	-1.264	0.000	1.000	-1.273	0.000
b2	1.182	1.000	1.227	1.136	1.273	0.000	0.000	0.455	0.000
f4	-0.655	0.000	-0.468	-0.641	-0.382	0.000	0.000	-1.136	1.000
	1.182	1.000	1.227	1.136	1.273	0.000	0.000	0.455	0.000

OK

Рис. 4

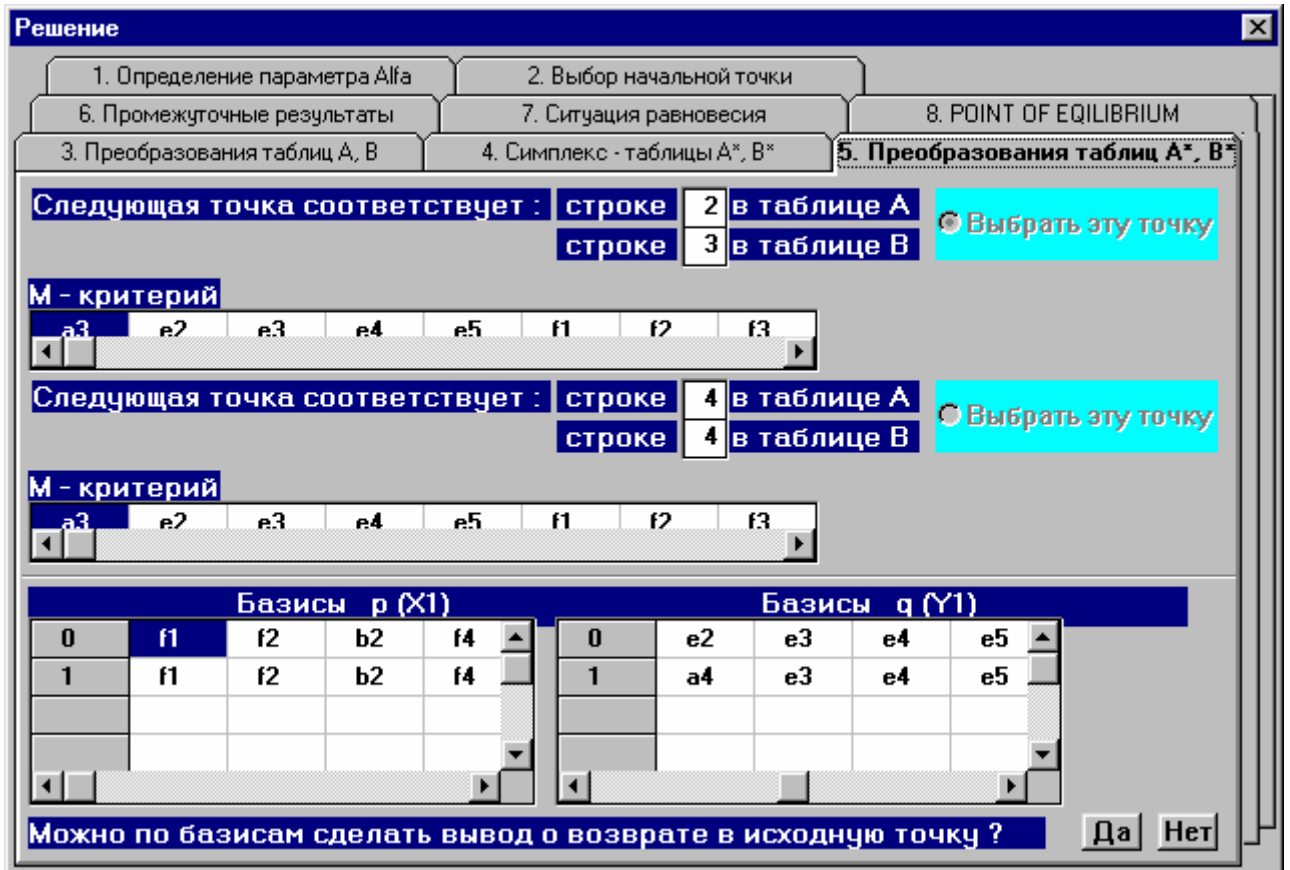


Рис. 5

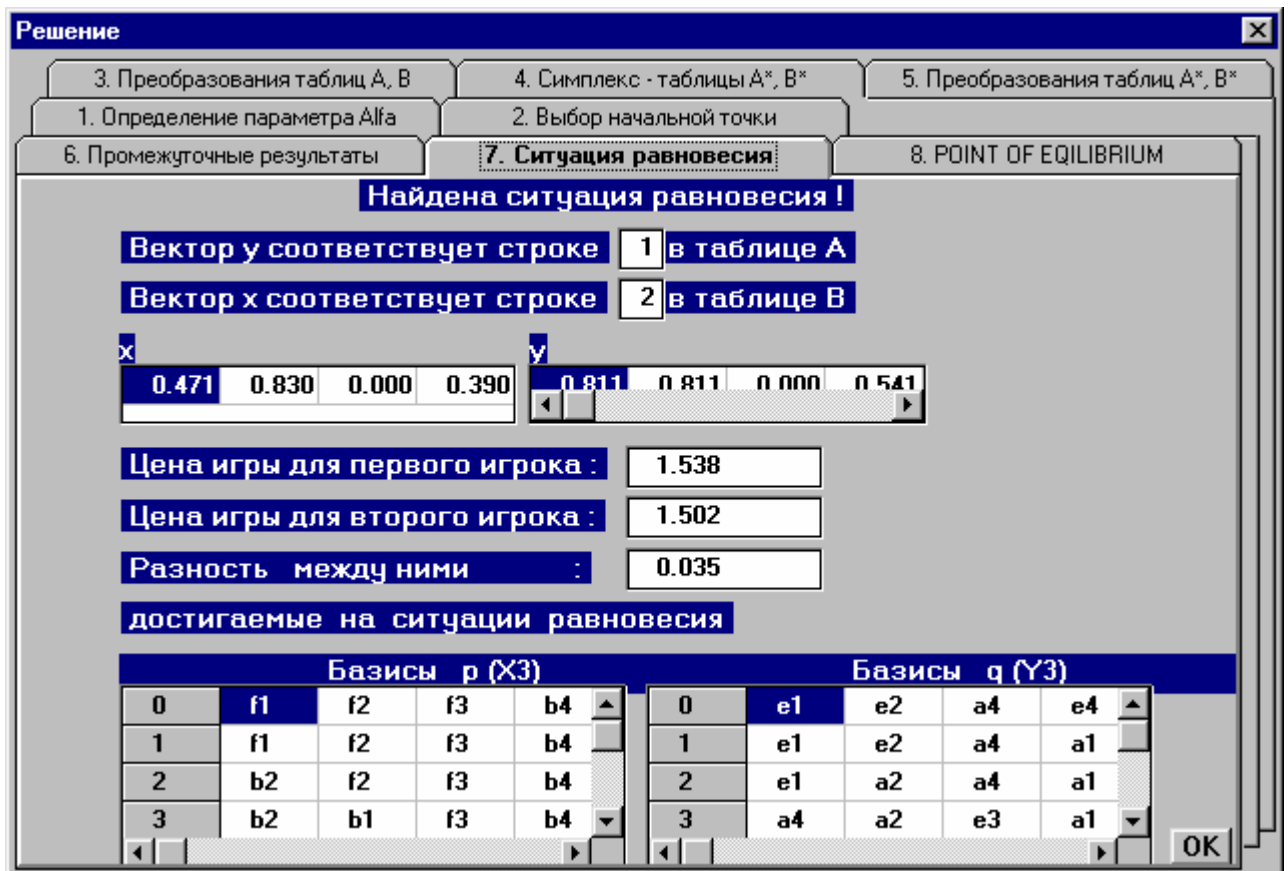


Рис.6

Вычисления по пути, определяемому по первой строке матрицы A , приводят обратно в начальную точку. Вычисления по новому S - пути, с другой начальной точкой, приводят к ситуации равновесия. Указываются значения цены игры и нормированные значения координат векторов, составляющих ситуацию равновесия. Вид соответствующего рабочего окна приведен на рис. 6.

Анализ примера показывает, что внедренный с помощью многооконной среды, контроль над ходом вычислений позволяет получить гарантированный результат. Участие пользователя в процессе принятия решений о выборе начальной и следующей точек позволяет избежать негативных особенностей алгоритма, такие как возврат с исходную точку, при неудачном выборе начального приближения.

Заключение

Внедрение подобных программных продуктов в процесс обучения, открывает широкие возможности для исследований свойств алгоритмов. Освобождая от необходимости проводить большой объем вычислений, такие программы позволяют провести исследования вопросов сходимости, возможных способов вычислений решения, зависимость числа шагов от выбора начальной точки и т.д. Также можно создавать программы для алгоритмов, которые не представлены в стандартном программном обеспечении, привлекая к такой работе самих студентов. Это хорошая практика для работы программиста с одной стороны и вычислительная практика для математиков с другой.

Библиографический список

1. Партсахаратхти Т., Рагхаван Т. Некоторые вопросы теории игр двух лиц. –М.,МИР, 1974. 295 стр.
2. Митчелл К. Керман. Программирование и отладка в Delphi. Учебный курс.- М., Вильямс, 2004. 720 стр.

Сведения об авторе.

Набатова Дария Сергеевна, доцент кафедры «Математическая кибернетика» Московского авиационного института (Государственный технический университет), кандидат физико-математических наук;

Москва, Волоколамское шоссе, д. 4

+7 499 158-48-11, nabatova805@mail.ru