каждой фирмы кодируется бинарной строкой длиной l. Эта строка кодирует действительное число, обозначающее количество продукции, которое решила произвести фирма i в цикле t.

Обозначим количество товаров, производимых фирмой i в цикле t через $y_{i,t}$. Лица, принимающие в фирме решения, не знают цены следующего цикла в том момент, когда они определяют выпуск фирмы. Но им необходимо иметь ожидаемую цену $p_{i,t}^{e}$. Исходя из этого ожидания, фирма i выбирает уровень выпуска, который принесет ей ожидаемой доход $\prod\left(p_{i,t}^{e}, y_{i,t}\right)$ настолько большим, насколько это возможно. Оптимальное количество продукции для фирмы i задается через $y^{*}\left(p_{i,t}^{e}\right)$. Цена, определяющая рынок в цикле t, обозначается через p_t и определяется с помощью обратной функции спроса. При рыночном равновесии ожидаемая цена должна в последствии совпадать с ценой равновесия, то есть $p_t = p_{i,t}^{e}$ для всех i, и все фирмы предпринимают одинаковые оптимальные действия $y^{*}\left(p_{i,t}^{e}\right) = y^{*}$ для всех i и для некоторых $y^{*} \in y^{*}(p_t)$.

Начальное состояние популяции задается случайным образом. Переход из цикла t в $t+1$ цикл осуществляется путем применения операторов генетического алгоритма [2]. Сначала необходимо представить стратегию каждой фирмы бинарной строкой. В работе количество выпускаемой продукции для k-ой фирмы определялось соотношением: $y(k) = \dfrac{ax}{bn}$, где $x = \sum_{i=1}^{l} 2^{-i} k(i)$, $k(i) \in \{0,1\}$ - значение i-го бита в строке k. Затем задавалась фитнес-функция, реализующая принцип выживания сильнейшего. В качестве фитнес-функции бралась функция дохода:

$$f_k(f) = \prod\left(p(f), y(k)\right),$$

где $p(f)$ означает цену, при условии, что вся популяция находится в состоянии $f \in S$.

Далее при помощи оператора пропорционального отбора выбирались фирмы, чьи значения функции приспособленности были большими. Тогда стратегии, дававшие среднюю прибыль, будут в дальнейшем в среднем использованы большим количеством агентов. Выбранные строки на следующем этапе разбиваются напополам, и к каждой паре с заданной вероятностью применяется оператор одноточечного кроссовера. После применения кроссовера каждый бит в каждой строке меняется на противоположный с заданной мутационной вероятностью.

Для моделирования была написана программа в пакете MATLAB при помощи инструментальной панели GA Toolbox. В данной инструментальной панели используются матричные функции пакета MATLAB для создания набора универсальных инструментов, обеспечивающих применение широкого диапазона методов генетических алгоритмов.

В результате проведения ряда экспериментов было обнаружено, что при любом задании параметров паутинообразной модели наблюдается резкое приближение к равновесному уровню. То есть фирмы способны были изменять свои решения таким образом, что количество выпускаемой ими продукции оставалось оптимальным, после довольно короткого периода эволюции.

В классической паутинообразной модели [3] возможна ситуация, когда на рынке не существует равновесия. Это происходит тогда, когда абсолютный наклон линии спроса превышает наклон линии предложения. Вместе с тем, эволюционное моделирование паутинообразной модели, проведенное в пакете MATLAB, демонстрирует, что процесс эволюции всегда сходится к некоторому равновесному состоянию вне зависимости от абсолютных наклонов линий спроса и предложения. Можно показать, что теорема шим (Schema theorem) объясняет данный феномен.

СПИСОК ЛИТЕРАТУРЫ

1. J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press: Ann Arbor, USA, 1975.

2. Herbert Dawid, Michael Kopel. On economic applications of the genetic algorithm: a model of the cobweb type. // Journal of Evolutionary Economics (1998), 8: 297-315.

3. Гальперин В.М., Игнатьев С.М., Моргунов В.И. Микроэкономика. Электронный учебник.

МЕТОД ПРЯМОГО СЧЕТА В ИССЛЕДОВАНИИ РЫНОЧНОЙ СИТУАЦИИ

Клёнов М.В.
Самарская государственная академия путей сообщения, Самара

Допустим мы имеем некоторую рыночную ситуацию в идеальном рынке. Для того чтобы найти наиболее вероятную таблицу расклада [4] предлагаю воспользоваться следующей логикой

У i-го предложения существует ниша, которую оно готово предоставить под j-ый спрос. У j-го спроса существует ниша, которая может реализоваться за счет i-го спроса. Это разные величины, однако практические $C_j^i = Д_i^j = \min(нишаC_j^i; нишаД_i^j)$.

Тогда для n-го уровня спроса и первого уровня предложения получаем:

$$C_1 \times \frac{Д_n}{\sum_{j=1}^{n} Д_j} \geq Д_1^n(C_n^1) \leq Д_n \times \frac{C_1}{\sum_{i=1}^{n} C_i},$$

а значит

$$Д_1^n(C_n^1) = \min(C_1 \times \frac{Д_n}{\sum_{j=1}^{n} Д_j}; Д_n \times \frac{C_1}{\sum_{i=1}^{n} C_i}) =$$

$$= \min(Д_n \times \frac{C_1}{\sum_{j=1}^{n} Д_j}; Д_n \times \frac{C_1}{\sum_{i=1}^{n} C_i})$$

Поскольку на различных этапах расчета какого-то одного Д$_j$ ($Д_j = \sum_{i=1}^{j} Д_i^j$)(верхний индекс – ценовой уровень спроса, нижний – ценовой уровень предложения, нишу в котором занимает данный спрос; у предложения наоборот) возможно использование различных частей формулы, то необходимо в последующих расчетах учитывать результаты предыдущих, устраняя их из расчетов. В противном случае смена формулы расчета min приведет к эффекту расчета по этой формуле всех предыдущих вариантов, а, значит, автоматически приведет к ошибке.

Поэтому для $Д_i^n(C_n^i)$ имеем

$$Д_i^n(C_n^i) = \min((Д_n - \sum_{k=0}^{i-1} Д_k^n)\frac{C_i}{\sum_{p=i}^{n} C_p};$$

$$(Д_n - \sum_{k=0}^{i-1} Д_k^n)\frac{C_i}{\sum_{p=i}^{n} Д_p - \sum_{k=0}^{i-1} Д_k^n})$$

Напомним, что нулевого уровня спроса, как и предложения не существует.

Расчет по этой формуле возможен для всех уровней, если после каждого расчета удалять n-ый уровень спроса и предложения, а из всех уровней предложения (от 1 до n-1) вычесть C_n^i. В итоге уровень n-1 станет уровнем n'.

Однако этот «окольный» вариант можно было бы получить лишь в случае разработки формул для более высокого уровня, поскольку они имеют ряд особенностей. Именно их разработка позволила впоследствии получить приведенную формулу для уровня n.

Для уровня n-1 формула будет иметь вид:

$$Д_i^{n-1}(C_{n-1}^i) = \min((Д_{n-1} - \sum_{k=0}^{i-1} Д_k^{n-1})\frac{C_i - C_n^i}{\sum_{p=i}^{n-1} C_p - \sum_{p=i}^{n-1} C_n^p};$$

$$(Д_{n-1} - \sum_{k=0}^{i-1} Д_k^{n-1})\frac{C_i - C_n^i}{\sum_{p=i}^{n-1} Д_p - \sum_{k=0}^{i-1} Д_k^{n-1}}$$

Соответственно для того, чтобы рассчитывать спрос для уровня n-1 нужно рассчитать спрос (и само собой разумеется предложение) для уровня n.

Соответственно для того, чтобы рассчитать спрос для j-го уровня требуется расчет спроса для уровней n ... j+1.

Для $Д_i^j(C_j^i)$, где j$\geq$i имеем формулу вида

$$Д_i^j(C_j^i) = \min((Д_j - \sum_{k=0}^{i-1} Д_k^j)\frac{C_i - \sum_{l=n}^{j+1} C_l^i}{\sum_{p=i}^{j} C_p - \sum_{p=i}^{j}\sum_{s=n}^{j+1} C_s^p};$$

$$(Д_j - \sum_{k=0}^{i-1} Д_k^j)\frac{C_i - \sum_{l=n}^{j+1} C_l^i}{\sum_{p=i}^{j} Д_p - \sum_{k=0}^{i-1} Д_k^j}$$

Соответственно, использование данной формулы для ситуации с пятью ценовыми уровнями спроса и предложения (столбец Д$_j$ и строка С$_i$ из табл. 1) выделило следующую наиболее вероятную таблицу расклада (представлены только сделки), которая размещена в строках и столбцах 1-5 в таблице 1.

Таблица 1. Результаты расчетов

		C_j^i							
	j \ i	1	2	3	4	5	Д$_j$	Д$_\Sigma$	Д$_{остаточное}$
$Д_i^j$	1	3	-	-	-	-	40	3	37
	2	2	8	-	-	-	35	10	25
	3	2	6	21	-	-	30	29	1
	4	2	4	7	9	-	22	22	-
	5	1	2	4	5	6	18	18	-
	С$_i$	10	20	32	38	49			
	С$_\Sigma$	10	20	32	14	6			
	С$_{остаточное}$	-	-	-	24	43			

СПИСОК ЛИТЕРАТУРЫ

1. Вентцель Е.С. Теория вероятностей. // М.: Высшая школа, 2002
2. Гмурман В.М. Теория вероятностей. Учебник для ВУЗов.// М.: Высшая школа, 2003
3. Евтодиева Т.Е. Логистические основы процесса сбытовой деятельности// Самара, СГЭА, 2000
4. Клёнов М.В., Ольшанский А.М. Моделирование сбыта продукции предприятия // Самара, 2004