УДК 681.3

ПРИМЕНЕНИЕ ЛОКАЛЬНОГО ЭВОЛЮЦИОННОГО АЛГОРИТМА В ПРОГНОЗИРОВАНИИ РАЗВИТИЯ ФИНАНСОВОГО РЫНКА

канд. техн. наук, доц. А.О. ГЛУХОВ,
Е.А. КИРШИН, А.В. ЮГАНОВ
(Полоцкий государственный университет)

Представлен локальный эволюционный алгоритм многомерной аппроксимации и его применение в финансовой сфере при прогнозировании курса акций и объема продаж за определенный промежуток времени.

Введение. В мире экономики огромное значение имеет проблема обработки статистической информации, которая используется аналитиками для оценки ситуации на рынке, анализа особенностей и тенденций изменения различных показателей рынка, таких как курс валют, стоимость акций, объем продаж и др. Обрабатывать огромные объемы информации с помощью традиционных методов достаточно сложно и неэффективно. Особенно остро стоит проблема прогнозирования развития рынка, ведь каждый просчёт может обернуться серьезными последствиями.

С развитием компьютерной техники появилась возможность применять программные средства для обработки статистических данных. Сейчас создано множество различных программных пакетов, специально разработанных для этой цели.

Самым популярным средством, на основе которого создаются программные комплексы обработки статистической информации являются нейронные сети.

Важнейшим свойством нейронных сетей является их способность к обучению. Обладая некоторой входной информацией, нейронные сети могут извлекать из нее скрытые закономерности, часто не обнаруживаемые традиционными методами корреляционного анализа. Применение нейронных сетей в финансовой сфере основано на следующем допущении - замене прогнозирования распознаванием [1]. Нейросеть не предсказывает будущее, она лишь пытается на основании полученных данных «узнать» (найти похожую) ранее встречавшуюся ситуацию и дать соответствующую реакцию на такое стечение обстоятельств.

Для получения наиболее точных результатов прогнозирования необходимо соответствующее обучение нейронной сети. Обучение нейронной сети сводится к аппроксимации зависимости между выходными и входными значениями. Однако нейронная сеть обладает ограничениями, не позволяющими аппроксимировать некоторые типы поверхностей [2].

Задача аппроксимации может быть решена несколькими методами [3,4]. Один из них - рассматриваемый нами локальный эволюционный алгоритм (ЛЭА).

В основе эволюционного алгоритма лежит биологическая модель эволюции. Как известно, в процессе эволюции выживают наиболее приспособленные особи. Это приводит к тому, что приспособленность популяции возрастает, позволяя ей лучше выживать в изменяющихся условиях. Поэтому сущность генетического алгоритма состоит в создании популяции, получении новых особей из существующих и замене «плохих» особей более приспособленными [3].

Особью в генетическом алгоритме является объект, кодирующий один из вариантов решения поставленной задачи. Например, в случае задачи аппроксимации объект может кодировать функциональную зависимость.

Эволюционные алгоритмы применяются в решении таких задач, как распознавание текста и звука, моделирование, нахождение корней уравнений, решение систем уравнений, и др. Однако основным недостатком генетического алгоритма является его эвристический характер, что означает отсутствие гарантии достижения приемлемого результата за определенный промежуток времени.

Популяция. В данном ЛЭА популяция представляет собой набор подпопуляций, упорядоченный по значениям качества аппроксимации лучших в них хромосом (рис. 1). Присутствие в популяции функций одного типа не допускается. Алгоритм считает две некоторые функции функциями одного типа, если они отличаются только константами.

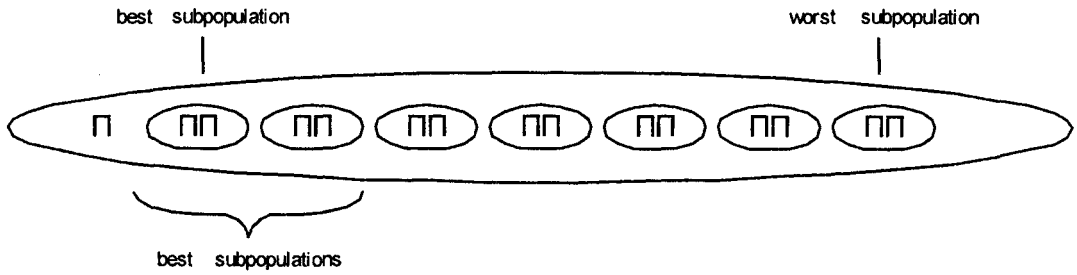

Рис. 1. Структура популяции

Размер популяции не является постоянным и изменяется в процессе удаления и добавления в нее хромосом, но всегда остаётся больше величины BestSubPopCount - числа лучших хромосом в популяции, которые никогда из неё не удаляются, так как дают лучшее качество аппроксимации во всей популяции, и предполагается, что в результате применения к ним различных эвристик с большей вероятностью получаются «хорошие» хромосомы, чем таковые получаются из более худших подпопуляций. Введение величины BestSubPopCount обусловлено также тем, что имеет смысл удаление из популяции «плохих» и, при этом переставших улучшаться подпопуляций, так как при этом происходит освобождение ресурсов процессора для обработки лучших подпопуляций.

Подпопуляция. Подпопуляция является набором хромосом, кодирующих функции одного типа (рис. 2).

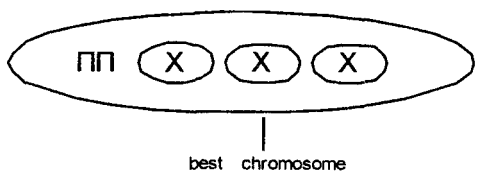

Рис. 2. Структура подпопуляции

Существование подпопуляций обусловлено тем, что в процессе улучшения констант функция может войти в локальный оптимум, т.е. константы в ней могут принять такие значения, изменение одного из которых ухудшает качество аппроксимации этой функции, при том, что функция данного типа может достичь лучшего качества аппроксимации. Это означает, что функция входит в локальную «яму» на поверхности $Q(c_1, c_2, \ldots, c_n)$, где Q – оценка качества аппроксимации (см. далее), n – количество констант в функции.

Частично проблема вхождения в локальный оптимум решается присутствием в популяции функций, отличающихся только константами, а также глобальными мутациями констант (см. далее). Для того, чтобы контролировать число функций одного типа в популяции и предназначены подпопуляции.

В структуре ПП следует выделить лучшую хромосому. Именно по ней и происходит оценка всей ПП при различных операциях с ней.

Хромосома. Как и в традиционных генетических алгоритмах, так и в разработанном ЛЭА элементами алгоритма являются хромосомы. Однако в данном алгоритме делается отступление от представления хромосомы как набора генов, т.е., если рассматривать программную реализацию в виде некоторой последовательности битов. Хромосомы в данном ЛЭА представлены в виде деревьев. Элементы дерева - это узлы трех типов: оператор, переменная либо константа. Каждая хромосома кодирует объекты поиска - функции. Деревья являются удобным способом представления функции, так как обеспечивают простоту осуществления некоторых необходимых эвристик (Grow, Crossover, Mutation - см. далее). В данном случае нет необходимости делать многочисленные проверки на правильность осуществления эвристики, что повышает скорость работы алгоритма.

Рассмотрим представление функции в виде дерева. Предположим, что функциональная зависимость имеет следующий вид:

$$f(X1, X2) = 3.26 \cdot X2 + \lg(X1 + \sqrt{X2 \cdot (-1.52)}).$$

Тогда дерево будет представлено так, как показано на рис. 3.

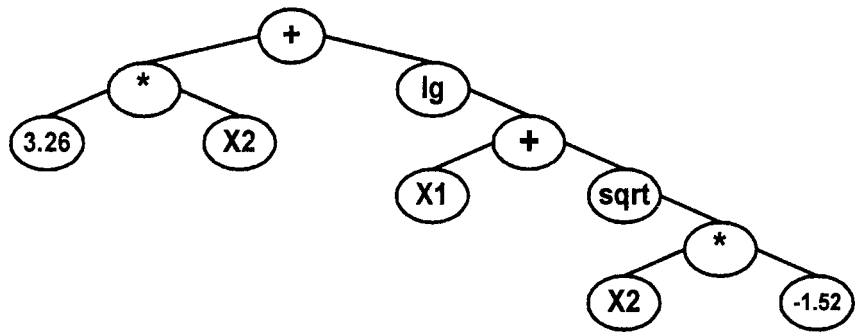

Рис. 3. Пример представления функции в виде дерева

Узлы дерева - операторы, листья - константы и переменные.

Глубина дерева хромосомы - наибольшее из значений уровней его узлов. В алгоритме глубина функции не может превышать заданного параметром MaxFuncDepth значения.

Для подсчета выходного значения функции необходима рекурсивная функция, считающая значение поддерева. Применяя последовательно эту функцию для каждого узла, начиная с вершины, получаем выходное значение.

При каждой функции есть добавочная константа, предназначенная для сдвига функции (см. далее). Значение этой константы прибавляется к выходному значению функции.

Операции. Данный алгоритм решает задачу аппроксимации путем поиска оптимальной функциональной зависимости для заданных входных замеров. Функции состоят из определенного набора операций, переменных и констант. Операции выбираются из заданного множества определенных операций. На данный момент в алгоритме применяются следующие операции: сложение; вычитание; умножение; деление; экспонента; десятичный логарифм; синус; квадратный корень.

В процессе работы программы возможен выбор операций, составляющий аппроксимирующую функцию. Кроме того, набор операций можно легко расширить: необходимо лишь описать обработчик добавляемой операции.

Оценка качества аппроксимации. Оценка качества аппроксимации *(Q)* - критерий, по которому идет отбор среди особей в эволюционном алгоритме. Для оценки величины Q служит фитнесс-функция - функция расчета «подходимости» функции под заданные замеры.

Замеры - это входные данные для алгоритма. Они представляют собой набор значений аргументов и соответствующих им значений функции. В данном алгоритме замеры загружаются из файла, где содержатся в следующем формате:

$$X_{11}\, X_{12}\, X_{13} \ldots X_{1N}\, Y_1$$
$$X_{21}\, X_{22}\, X_{23} \ldots X_{2N}\, Y_2$$
$$\cdot \quad \cdot \quad \cdot$$
$$X_{M1}\, X_{M2}\, X_{M3} \ldots X_{MN}\, Y_M,$$

где X_{ij} $(1 \le i \le M, 1 \le j \le N)$ – значения аргументов; Y_i – значения функции; M – число замеров; N – число переменных.

Критерий подходимости Q рассчитывается как среднеквадратичное отклонение рассчитанных значений функции в точках замеров от заданных значений функции в точках замеров:

$$Q = \frac{\sum_{i=1}^{N}(F(X_{1i},\ldots,X_{Mi})-Y_i)^2}{N} = \frac{\sum_{i=1}^{N}\Delta F_i^2}{N}.$$

Для ускорения нахождения аппроксимации в алгоритм внедрена операция сдвига функции (ShiftFunc).

При каждой хромосоме имеется добавочная (сдвигающая) константа Const, которая прибавляется к функции:

$$F = f + Const,$$

где f – значение, посчитанное по дереву; F – результирующее (выходное) значение.

Добавочная константа обеспечивает сдвиг вдоль оси функции. В алгоритме эта константа необходима для того, чтобы сместить функцию вдоль этой оси в оптимальное положение.

Значение сдвигающей константы вычисляется как среднее отклонение рассчитанных значений функции в точках замеров от заданных значений функции в точках замеров со знаком «минус»:

$$Const = -\frac{\sum_{i=1}^{N}(f(X_{1i},...,X_{Mi})-Y_i)}{N} = -\frac{\sum_{i=1}^{N}\Delta f_i}{N},$$

где $X_{1i},..., X_{Mi}$ – значения переменных в точках замеров; M – количество переменных; N – количество замеров; Yi – значения замеров.

Основные эвристики. Эвристика в генетическом алгоритме - это действие, в результате которого может быть сделано улучшение текущей популяции особей. Эвристики могут выполняться как над особями, так и над всей популяцией в целом.

Эвристики являются неотъемлемой частью генетического алгоритма, потому что именно они создают новые особи на основе старых, т.е. являются источником хромосом, обновляющих популяцию - источником нового поколения.

Хорошо подобранный набор эвристик определяет эффективность генетического алгоритма в целом. Необходимо отметить, что на эффективность также влияет и соотношение процента применения эвристик.

Набор эвристик данного алгоритма следующий: рандомная (произвольная) генерация, мутация узлов, мутация констант, доращивание, скрещивание, сдвиг функции, расширение.

Эвристики можно классифицировать следующим образом:

1. Эвристики воздействия на особей (рандомная (произвольная) генерация, мутация узлов, мутация констант, доращивание, скрещивание, сдвиг функции);
2. Эвристики, применяемые ко всей популяции (расширение);

Эвристики воздействия на особей можно классифицировать по следующим признакам:

1. По масштабу изменения функции:
   а) локальные (мутация узлов, локальная мутация констант, доращивание, скрещивание);
   б) глобальные (рандомная генерация, глобальная мутация констант).
2. По типу изменяемых узлов:
   а) изменяющие только константы (мутация констант);
   б) изменяющие любые узлы (рандомная генерация, мутация узлов, доращивание, скрещивание).

Необходимо отметить, что выбор узла для изменения осуществляется случайным образом, но учитывая необходимые ограничения.

Локально-эволюционный алгоритм

1. Создание начальной популяции

{// цикл по подпопуляциям в популяции

{ // цикл по хромосомам в подпопуляциях

а) получение новой (новых) хромосомы (хромосом) из текущей;

б) если новая хромосома проходит критерий добавления в популяцию (NewChrom-> Q / WorstChrom-> Q <= AddingCriterion), то на её основании создаётся подпопуляция, которая добавляется в популяцию, вытесняя при этом самую худшую подпопуляцию, если популяция имеет максимальный размер. Однако в том случае, когда хромосома, кодирующая функцию такого же типа, в популяции уже существует, эти действия не производятся;

в) мутация констант у текущей хромосомы. Если новая хромосома предпочтительнее старой, то она заменяет последнюю.}

2. Проверка текущей подпопуляции на возможность её удаления. Если в течение MaxBadCyclesCount циклов данная подпопуляция не улучшилась значительно и она не входит в число BestSubPopCount лучших подпопуляций, то она удаляется из популяции.

3. Изменение критерия добавления. Если в течение MaxNonAddingCyclesCount циклов ни одна функция не была добавлена в популяцию, то критерий добавления AddingCriterion увеличивается на величину AddingCriterionChangeStep, иначе он уменьшается на эту же величину, оставаясь при этом больше величины MinAddingCriterion.

4. Проверка на критерий расширения. Если в течение ExpandCriterion циклов лучшее Q в популяции не уменьшилось, то число подпопуляций в популяции увеличивается до начальной величины StartSubPopCount за счёт добавления случайно сгенерированных хромосом.

5. Удаление из популяции «лишних» подпопуляций, т.е. если размер популяции больше величины MaxSubPopCount, то он постепенно уменьшается до этой величины. }

Если на каком-либо этапе полученная новая хромосома удовлетворяет критерию завершения работы Q < FinishCriterion, то алгоритм прекращает работу и выдаёт найденный результат.

Следует отметить, что существование параметра AddingCriterion обусловлено тем, что в процессе работы алгоритма может сложиться ситуация, когда при AddingCriterion = 1 новые хромосомы перестанут добавляться в популяцию. Это произойдёт, когда популяция улучшится значительно по сравнению со своим начальным состоянием и поэтому вероятность возникновения новой особи, которая была бы лучше самой худшей хромосомы в популяции, будет мала. Увеличение этого параметра в случае длительного недобавления в популяцию новых особей позволяет алгоритму не заходить в тупик, а продолжать добавлять в популяцию новые, более худшие особи, которые, тем не менее, могут улучшиться и дать хороший результат.

Применение алгоритма на практике. Пусть необходимо на основании имеющихся статистических данных по курсу акций IBM дать прогноз на ближайшее будущее. Значения курса даны на определенный срок по неделям. Выбранный метод составления прогноза - аппроксимация зависимости между выходной переменной - значением курса за пятую неделю и входной - значением курса за последние четыре недели [5].

Данный таким образом прогноз предполагает изменение всех внешних факторов, влияющих на курс акций таким же образом, каким эти факторы изменялись во время такой же ситуации в прошлом. Очевидно, что такой прогноз не является точным и может быть использован лишь опытными финансовьми аналитиками, способными оценить влияние внешних факторов. В рассматриваемом примере влияние внешних факторов не учитывается. Рассматривается лишь закономерность изменения значения продаж.

Сравним качество аппроксимации, полученное с помощью локального эволюционного алгоритма и нейронной сети. Результаты сравнения представлены на рис. 4.

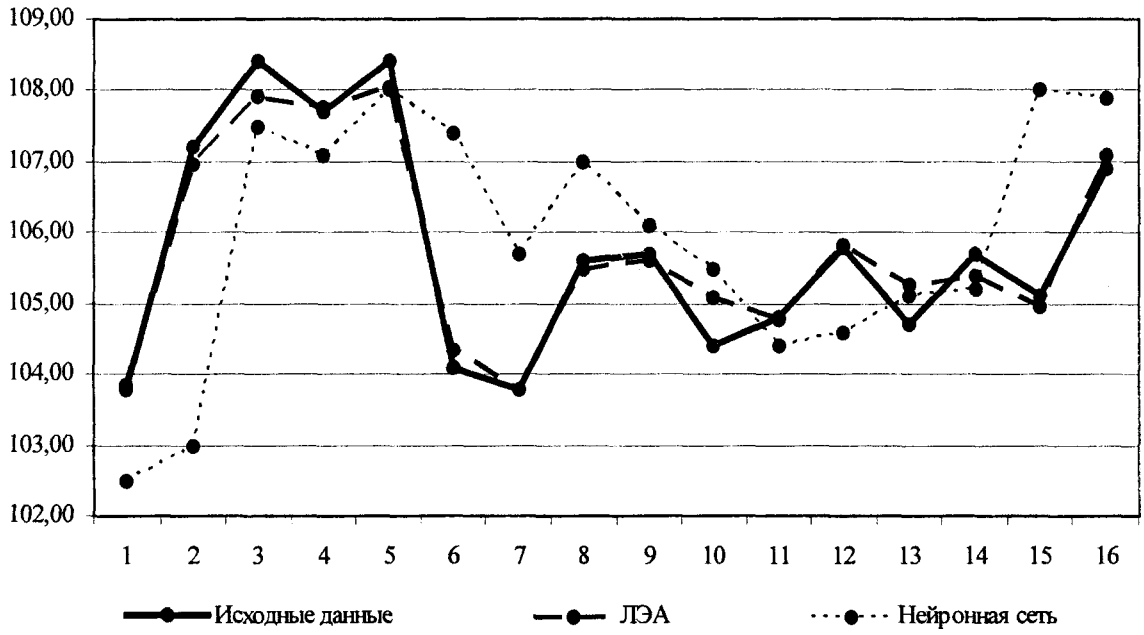

Рис. 4. Сравнение аппроксимации ЛЭА и нейронной сети

Из графика видно, что ЛЭА дает лучшее качество аппроксимации, чем нейронная сеть.

Аппроксимируем, например, зависимость индекса фондовой биржи NASDAQ от индекса предыдущих шести недель за период с 7 апреля по 24 августа 2003 года. На рис. 5 видно, что ЛЭА даёт очень хорошее качество аппроксимации.

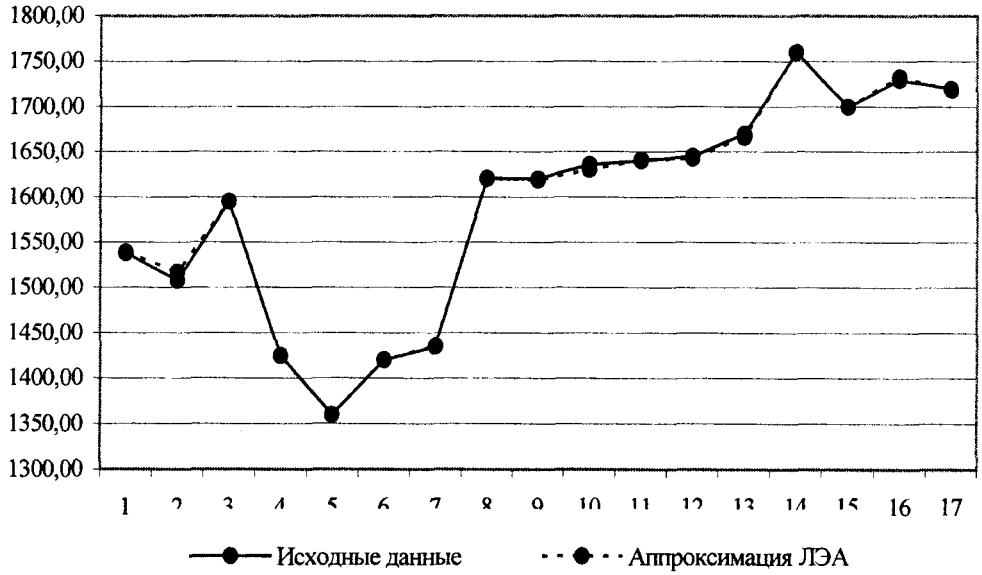

Рис. 5. Аппроксимация зависимости индекса фондовой биржи NASDAQ
от индекса предыдущих шести недель за период с 7 апреля по 24 августа 2003 года

Заключение. Таким образом, локальный эволюционный алгоритм способен дать лучшее решение задачи аппроксимации, чем нейронная сеть. А так как прогнозирование в менеджменте сводится к аппроксимации зависимости, то можно сделать вывод, что прогнозирование с применением ЛЭА также будет более точным. Главной задачей при этом является построение правильной аппроксимационной модели, способной в достаточной мере выявить закономерности развития ситуации на рынке.

ЛИТЕРАТУРА

1. Блощицкий А. Исследование средств финансового прогнозирования на основе нейронных сетей // URL: http://masters.donntu.edu.ua/2000/fkita/bloshits/nauka.html
2. Уоссермен Ф. Нейрокомпьютерная техника. - М: Мир, 1992. - 184 с.
3. Минаков И.А. Сравнительный анализ некоторых методов случайного поиска и оптимизации // Известия Самарского научного центра Российской академии наук. - 1999. - № 2.
4. Зайцева Е.Н., Станкевич Ю.А. Некоторые современные методы решения оптимизационных задач // Новые информационные технологии в образовании: Материалы Второй междунар. конф., -1996.
5. Некипелов Н. Опыт прогнозирования финансовых рынков // URL: http://www.basegroup.ru/neural/stockmaiket.htm