



## EVOLUTIONARY ALGORITHMS FOR ALGORITHMIC TRADING ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ ДЛЯ АЛГОРИТМИЧЕСКОЙ БИРЖЕВОЙ ТОРГОВЛИ

**Cholak K.U. / Чолак К.Ю.**

*Kharkiv National University of Radio Electronics, Kharkiv, Nauky Ave, 14, 61000*

*Харьковский национальный университет радиоэлектроники,*

*Харьков, пр. Науки, 14, 61000*

**Аннотация:** Рассмотрено использование алгоритма эволюционного обучения как промежуточного шага в использовании нейросетей для алгоритмической торговли. Эволюционный алгоритм позволяет решить проблему разметки данных для обучения. Правильная разметка тренировочных данных значительно повышает шансы успеха при поиске зависимости между рыночной ситуацией и оптимальной настройкой. А полная автоматизация генерации разметок позволяет алгоритму адаптироваться под изменения рынка. Склонность эволюционных алгоритмов к перенасыщению позволяет находить оптимальные, хоть и не общие решения, то есть наилучшие решения для конкретной рыночной ситуации, непереносимые на другие ситуации.

**Ключевые слова:** Эволюционные алгоритмы, торговые алгоритмы, искусственные нейронные сети, перенасыщение.

### **Вступление**

Так как зачастую алгоритмы торговли являются нетривиальными и требуют тонкую настройку, то человеку сложно определить оптимальный вариант настройки в каждый момент времени для максимизации желаемого показателя. С задачей максимизации отлично справляются эволюционные алгоритмы, к тому же они не требуют разметки данных, они самостоятельно определяют какое поведение является оптимальным.

Большой проблемой на пути использования эволюционного алгоритма как самостоятельного элемента при поиске генерализованного подхода к настройке торговых алгоритмов является склонность к перенасыщению, то есть заучиванию данных. Алгоритм эволюционного обучения отлично справится с предоставленной ему задачей, но не будет способен решать эту же задачу на данных, которые он ранее не видел. Из-за чего вместе с эволюционным алгоритмом возникает необходимость использовать нейронные сети. Получив от эволюционного алгоритма оптимальный вариант настройки торговой системы за длительный период времени, нейросеть способна найти закономерность между оптимальными настройками и рыночными данными, при наличии этой закономерности в первую очередь.

### **Основной текст**

Алгоритмы эволюционного обучения построены на отборе агентов с наилучшими показателями, которые измеряются целевой функцией. Целевая функция может возвращать как один, так и множество показателей. В случае с множественной максимизацией обычно ищут оптимум по Парето. Следующим шагом алгоритма является создание популяции, каждый агент в популяции имеет случайным образом установленные веса, наподобие искусственной нейронной сети. На вход каждому агенту подаются тренировочные данные, которые представляют из себя текущую рыночную ситуацию, а на выход,



преобразовав входные данные с помощью весов, агент выдает торговые решения или набор настроек для торговой стратегии.

В случае, если агент выдает торговые решения, то он сам определяет стратегию. Это дает гораздо больше свободы агенту, что является проблемой, ведь найти закономерности становится гораздо сложнее, в отличие от варианта когда агент лишь может настраивать определенную экспертом торговую стратегию. Ведь действия стратегии значительно более предсказуемые, как и влияние настроек на ее поведение, что позволяет гораздо проще найти зависимость между рыночной ситуацией и оптимальными настройками торговой стратегии. После создания популяции со случайными весами и получения от каждого агента настроек для стратегии нужно оценить оптимальность выбранных настроек с помощью целевой функции. Чем выше целевая функция у каждого агента, тем больше вероятность, что агент останется жив для повторной итерации. Если агент умирает, то он заменяется копией другого, более успешного (с большим значением целевой функции) агента.

После этапа отбора начинается процесс мутации и пересечения популяции. Его целью служит создание разнообразия в выборке, для проверки большего количества вариантов настроек стратегии близкого к наилучшему найденному на данный момент агентами. Для этого веса некоторых агентов случайным образом слегка изменяются (мутации), либо у пары агентов частично перемешиваются веса (пересечение, crossover).

Желательно никак не изменять несколько наилучших агентов, чтобы случайно не испортить самые удачные решения, этот подход называется элитизмом.

Повторно отбирая наилучших агентов и постоянно привнося достаточно разнообразия в выборку агенты будут приближаться к оптимальным настройкам. Эволюционные алгоритмы отлично справляются с нетривиальными задачами максимизации, хоть они и требуют больше вычислений, чем алгоритмы градиентного спуска, но способны найти более значительно оптимальное решение, так как градиентный спуск может застрять на локальном минимуме.

Так же нейронные сети требуют размеченные данные, то есть данные с указанными правильными ответами (классами, решениями, тд.), что часто невозможно сделать вручную либо потребует значительных усилий и экспертного анализа по конкретной алгоритмической стратегии, которого может попросту не быть на этапе разработке стратегии.

Обе проблемы - остановка на локальном минимуме и необходимость размечать данные решают алгоритмы эволюционного обучения. Остановка на локальном минимуме решается разбиением популяции на группы, где отбор наилучших и пересечение происходит только внутри группы, тем самым агенты получают возможность исследовать области за локальным минимумом, ведь агенты в локальном минимуме не будут снижать их шансы на выживание.

Группы, показывающие себя хуже других на протяжении нескольких поколений желательно заменять новыми, случайно созданными. Тем самым



получается вложенный эволюционный отбор - группы соревнуются между собой, как и участники каждой из них соревнуются внутри своей группы.

Проблема необходимости размечать тестовые данные для эволюционного алгоритма не возникает, так как в разметке нет необходимости.

Но эволюционному алгоритму свойственная своя проблема - перенасыщение. Алгоритм отлично справляется на предоставленной ему выборке данных, но не справляется с обобщением решения и проявляет себя плохо на новых для него данных. Эта проблема является активно изучаемой на момент написания статьи и общих принципов решения не существует. Вместо того, чтобы искать решение проблеме данная работа предлагает использовать результаты с перенасыщением как оптимальную настройку, которую можно использовать для разметки тестовых данных для дальнейшего обучения с помощью искусственной нейросети.

### **Заключение и вывод**

Был предложен и рассмотрен вариант использования алгоритмов эволюционного обучения с допущением перенасыщения как способ генерации оптимального решения задачи настройки произвольного торгового алгоритма. Найденное решение не будет генерализованным, но может быть использовано как разметка для нейросети. В сфере алгоритмической торговли применение искусственных нейронных сетей осложнено необходимостью разметки данных для обучения, указывая какое действие от нейросети ожидалось при заданных обстоятельствах. Разметка данных для обучения сложная задача для человека, так как нельзя с уверенностью сказать, что в какой момент нужно было делать для нетривиального торгового алгоритма из-за большого варианта развития событий. Задачу разметки тестовых данных можно решить с помощью алгоритма эволюционного обучения

***Abstract:** This work is studying automated parameterizing of handcrafted trading strategies and proposes the usage of evolutionary algorithms (EA) for labeling training data for neural networks (NN). NN alone are quite hard to use for controlling and parameterizing non-trivial trading strategies due to the need of supplying them with labeled training data. On the other hand EA are prone to overfit, often producing near to perfect results, without any generalization, thus these results can't be directly used in the future. But combining EA and NN we give NN an ability to generalize near to perfect decisions made by EA.*

***Keywords:** Evolutionary algorithms, trading algorithms, artificial neural networks, overfitting.*