

Министерство образования Республики Беларусь

Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.021

Хлопцев
Андрей Алексеевич

Генетический алгоритм для оптимизации настроек торговых ботов

АВТОРЕФЕРАТ

на соискание степени магистра

по специальности 1-40 80 04 – Информатика и технологии программирования

Научный руководитель
Сиротко С. И.
к.ф-м.н., доцент

Минск 2022

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

Торговый бот (советник, стратегия) – программный алгоритм, который выполняет сделки на бирже в автоматическом режиме по заранее определённым правилам.

МА – торговая стратегия, основанная на скользящем среднем (Moving Average).

ГА – генетический алгоритм.

Популяция (население, поколение) — подмножество всех возможных (закодированных) решений задачи поиска ГА.

Хромосома (особь, решение) – одно из решений задачи поиска ГА.

Ген – один из элементов хромосомы.

Аллель – значение, которое ген принимает для конкретной хромосомы.

Функция приспособленности – функция, принимающая хромосому в качестве аргумента, и выдающая пригодность решения в качестве выходных данных.

Генотип – популяция в вычислительном пространстве. Эти популяции представлены так, чтобы их можно легко понять и обрабатывать с помощью вычислительной системы.

Фенотип – популяция в реальном мире. Эти совокупность популяций в том виде, в каком они существуют в реальном мире.

Альфа-особь – вариант решения задачи, который относится к локальному максимуму, не являясь при этом оптимальным решением задачи. Может являться доминантным в популяции суживая область поиска оптимального решения.

Карта миграции – список правил, по которым острова генетического алгоритма обмениваются лучшими хромосомами. Обычно представляется в виде графа.

ВВЕДЕНИЕ

С древнейших времён торговля является одним из основных родов экономической деятельности, направленной на обмен товарами между людьми. На сегодняшний день биржевая торговля является одной из крупнейших отраслей торговли, которая непрерывно растёт и развивается. Например, только за 2021 год Нью-Йоркская торговая биржа (NYSE) совершила сделок на 26 трлн. долларов США.

С развитием биржевой торговли развиваются и способы совершения сделок, так, в последнее десятилетие, широкую популярность приобрела автоматизированная торговля. В данном виде торговли все сделки совершает не человек-брокер, а специальный написанный алгоритм. Однако, перед тем, как доверить свои деньги неодушевлённому алгоритму и запустить его для работы на торговых серверах, его необходимо правильно настроить. Т.к. результат торговли бота зависит от множества параметров, то данная задача может стать довольно нетривиальной, потому что необходимо перебрать огромное количество комбинаций значений параметров и учесть множество факторов и связей между ними.

Поэтому люди решили использовать специальные компьютерные алгоритмы, чтобы автоматизировать процесс поиска оптимальных настроек и свести шанс возникновения ошибки к минимуму. Самый очевидный вариант для получения таких настроек является алгоритм полного перебора (Brute force). [2] Хотя данный алгоритм всегда находит самую точную комбинацию настроек бота, время работы его очень велико, т.к. алгоритму необходимо перебрать все возможные комбинации параметров стратегии. Чтобы решить эту проблему, программисты решили использовать другие алгоритмы поиска решения, а именно эвристические алгоритмы поиска. [3] Данные виды алгоритмов находят не такое точное решение, как алгоритм полного перебора, однако работают значительно быстрее. В данной диссертации будет рассмотрен один из таких алгоритмов – генетический алгоритм, а также способы его адаптации к задаче поиска оптимальных настроек торговой стратегии.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Цель диссертации – разработка генетического алгоритма с адаптацией для поиска оптимальных вариантов настроек торговых ботов. Также алгоритм необходимо спроектировать таким образом, чтобы его запуск требовал минимальных знаний и действий от конечного пользователя при этом сохраняя баланс между средним временем работы алгоритма и точностью найденного решения. После разработки и тестирования алгоритм будет интегрирован в семейство программных продуктов TickTrader компании SoftFX.

Объект исследования – задача поиска оптимальных настроек торговых стратегий.

Предмет исследования – эвристические алгоритмы поиска для решения задач оптимизации, а в данном случае, генетический алгоритм, адаптированный под задачу поиска оптимальных настроек торговых стратегий.

Для достижения поставленной задачи необходимо:

- 1 изучить доменную область;
- 2 рассмотреть аналоги программных продуктов, присутствующие на рынке;
- 3 проанализировать отчёты и отзывы пользователей;
- 4 провести математический анализ поставленной задачи для её адаптации к ГА;
- 5 исследовать поставленную задачу для выбора лучших вариантов генетического алгоритма;
- 6 создать адаптированный ГА;
- 7 провести тестирование полученного алгоритма;
- 8 провести ряд исследований и измерений для оптимизации скорости работы алгоритма с сохранением точности результатов;
- 9 интегрировать ГА в конечный программный продукт.

Интеграция результатов исследования

Результаты диссертации были интегрированы в продукты TickTrader AlgoStudio и TickTrader WinTerminal компании ООО «Софт Эф Икс Дев» и выпущены в релиз.

Личный вклад соискателя

В процессе написания диссертации соискатель проанализировал предметную область, изучил продукты конкурентных компаний, отзывы и отчёты об ошибках пользователей продукта TickTrader AlgoStudio, провёл ряд

исследований и измерений в данной предметной области, создал, протестировал и интегрировал ГА в программные продукты.

Публикации результатов диссертации

По теме диссертации было опубликовано 3 работы в сборниках трудов и материалов международных конференций.

Структура и объём диссертации

Диссертация состоит из:

- 1 общей характеристики работы;
- 2 введения;
- 3 четырёх основных глав:

3.1 Глава 1 даёт краткое представление о доменной области автоматизированной торговли и описание ГА.

3.2 Глава 2 – получение математического представления задачи поиска оптимальных настроек торговых ботов с помощью ГА.

3.3 Глава 3 – программная архитектура ГА.

3.4 Глава 4, в которой приведены результаты работы ГА на различных задачах.

- 4 заключения;
- 5 библиографического списка;
- 6 приложения к диссертации.

Общий объём страниц диссертации – 75, объём основного текста – 57 страницы, количество рисунков – 34, количество таблиц – 6, количество использованных источников – 34.

ОСНОВНАЯ ТЕОРИЯ

Автоматизированная торговля

В современном мире движение торгового ордера от трейдера через брокера на биржу полностью компьютеризировано, т.к. компьютерная техника является практически неотъемлемой частью жизни человека. Всё больше программного обеспечения появляется на рынке, в том числе и ПО в сфере финансовых технологий.

Программное обеспечение в сфере финтех призвано облегчить повседневную работу трейдера в виде принятия торговых решений. Одним из инструментов, призванных помочь трейдеру есть торговые советники (боты), которые принимают решения за трейдера автоматически.

Работа с торговым советником на валютном рынке заключается в том, что на торговую платформу устанавливается специально разработанная программа, основную часть ценности которой составляет торговый алгоритм. Он программируется на проведение торговых операций на валютном рынке, в зависимости от текущих рыночных условий в соответствии с торговой стратегией, на основе которой писался торговый алгоритм. Все управление работой торгового советника также заранее прописано в программном коде, что позволяет автоматизировать все ручное управление инструментами валютного рынка.

Автоматизированная торговля на сегодняшний день набирает обороты, в связи с наличием огромного количества преимуществ таких как:

- компенсация человеческого фактора (влияния эмоций, усталости, когнитивных искажений);
- недоступная человеку скорость операций;
- функционирование 24/7;
- моментальное реагирование на изменение ситуации на рынке;
- одновременная работа с любым запрограммированным числом валютных пар и бирж;
- возможность тестирования торговых стратегий для их проверки на жизнеспособность.

Торговая стратегия для тестирования ГА

Для тестирования генетического алгоритма необходимо выбрать наглядную и простую торговую стратегию для получения и сравнения результатов работы ГА с другими алгоритмами. В качестве такого алгоритма был выбран Moving Average bot.

Торговая стратегия МА состоит из 4 параметров настройки:

- **MaximumRisk** – параметр, необходимый для того, чтобы рассчитать максимальный объём лота, который выставляет бот. Задаётся в процентах, вычисляется исходя из оставшихся свободных денег у трейдера на балансе. По умолчанию имеет значение 0.02 – это значит, что размер открытой позиции не может превышать 2% депозита.

- **DecreaseFactor** – параметр, ограничивающий убытки трейдера. По умолчанию имеет значение 3 – это означает, что после второй подряд убыточной сделки, бот уменьшит объём лота в **DecreaseFactor** раз от предыдущего объёма сделки, но не менее минимально разрешённого объёма сделки.

- **MovingPeriod** – параметр, для периода индикатора МА. По умолчанию имеет значение 12.

- **MovingShift** – параметр, отвечающий за сдвиг скользящей средней. По умолчанию имеет значение 6.

Бот МА осуществляет свою работу на основе данных технического индикатора *moving average*. Этот индикатор показывает среднее значение цены инструмента за некоторый период времени (1).

$$SMA_t = \frac{1}{n} \sum_{i=0}^{n-1} p_{t-i} = \frac{p_t + p_{t-1} + \dots + p_{t-i} + \dots + p_{t-n+2} + p_{t-n+1}}{n}, \quad (1)$$

где

- n – значение параметра **MovingPeriod**;
- p – цена закрытия i -го бара;
- t – значение параметра **MovingShift**.

На старте бот МА создаёт технический индикатор МА, передавая в него параметры **MovingPeriod** и **MovingShift**. В течение работы бота одновременно может быть открыта только одна торговая сделка.

Алгоритм работы бота МА следующий:

- 1 ожидание сигнала от терминала о закрытии бара;
- 2 расчёт последнего значения технического индикатора МА;
- 3 бот сравнивает рассчитанные значения с ценой открытия и закрытия бара. Если значение индикатора не попадает в промежуток между ценами открытия и закрытия бара, то возвращаемся на шаг 1. Иначе переходим к следующему шагу;
- 4 проверка количества открытых позиций. Если уже есть открытая позиция, то бот закрывает её;
- 5 бот открывает новую позицию;
- 6 возвращаемся на шаг 1.

Область поиска оптимальной настройки торговой стратегии

В качестве основной тестовой области поиска оптимальных настроек возьмём результат работы бота МА на нескольких наборах исторических данных.

Для их визуализации в трёхмерном пространстве переберём первые два параметра МА бота. Параметр MovingPeriod будет осью x, параметр MovingShift будет осью y, а параметр z – это результат торговли бота с соответствующими настройками.

В качестве иллюстрационного примера были взяты данные работы МА бота по символу EURUSD с периодом в 1 минуту за промежуток с 12.01.2021 по 05.01.2022. Начальный баланс аккаунта \$10000. Поиск осуществляется на параметрах MovingPeriod=[1..100] и MovingShift=[1..100], шаг изменения равен 1. Общее количество возможных комбинаций настроек равно 10000 (рисунки 1 и 2).

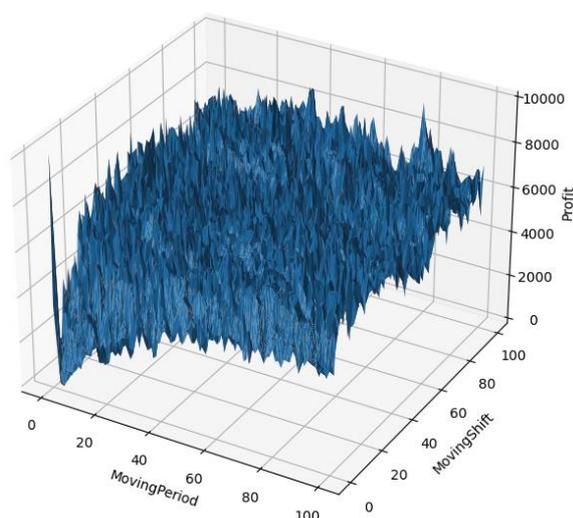


Рисунок 1 – Рельеф области поиска, построенный по символу EURUSD с помощью алгоритма полного перебора

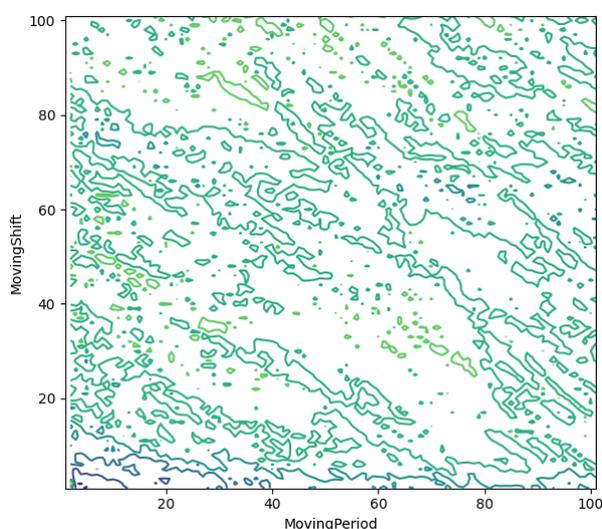


Рисунок 2 – Линии уровня функции МА на символе EURUSD

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Генетические алгоритмы стали особенно популярны благодаря работе Джона Холланда в начале 70-х годов и его книге «Адаптация в естественных и искусственных системах» (1975).

Поскольку алгоритм самообучающийся, то спектр применения крайне широк:

- задачи на графы;
- задачи компоновки;
- составление расписаний;
- создание «Искусственного интеллекта».

Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора («генотипа») генов, где каждый ген может быть битом, числом или неким другим объектом.

Некоторым, обычно случайным, образом создаётся множество генотипов начальной популяции. Они оцениваются с использованием «функции приспособленности», в результате чего с каждым генотипом ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо фенотип, им описываемый, решает поставленную задачу.

При выборе «функции приспособленности» важно следить, чтобы её «рельеф» был «гладким».

Из полученного множества решений («поколения») с учётом значения «приспособленности» выбираются решения (обычно лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» (в большинстве случаев «скрещивание» и «мутация»), результатом чего является получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение.

Этот набор действий повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма. Таким критерием может быть:

- нахождение глобального, либо субоптимального решения;
- исчерпание числа поколений, отпущенных на эволюцию;
- исчерпание времени, отпущенного на эволюцию.

В базовой реализации ГА можно выделить следующие этапы (рисунок 3):

- определение целевой функции (функции приспособленности) для особей популяции;

- создание начальной популяции;
- начало эволюционного цикла:
 - 1 размножение (скрещивание);
 - 2 вычислить значение целевой функции для всех особей;
 - 3 формирование нового поколения (селекция);

- 4 мутация генов;
- 5 если выполняются условия остановки, то (конец цикла), иначе (начало цикла). [9]

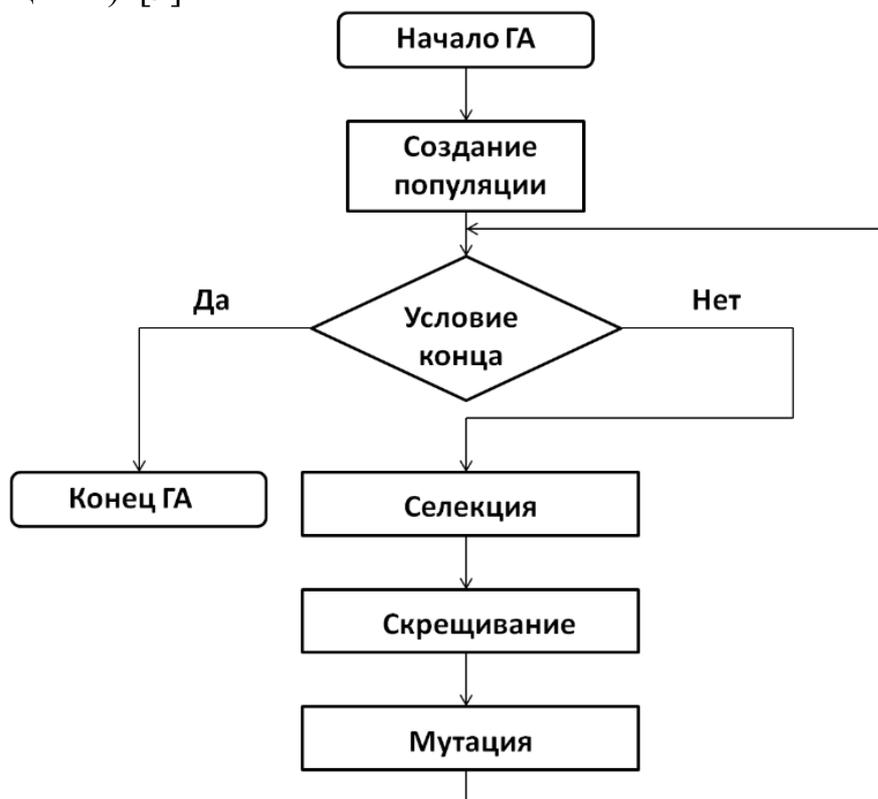


Рисунок 3 – Блок-схема базовой реализации генетического алгоритма

Асимптотику базовой версии ГА можно рассчитать по формуле (2):

$$O(n, gCount) = gCount \cdot n \cdot \log(n), \quad (2)$$

где

- $gCount$ – количество поколений;
- n – количество элементов в поколении;

МАТЕМАТИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ЗАДАЧИ ПОИСКА

Целевая функция задачи поиска оптимальной настройки торгового бота имеет вид:

$$\begin{cases} f(x_1, \dots, x_n) \rightarrow \sup; \\ \min_i \leq x_i \leq \max_i; \\ x_i \bmod \text{step}_i = 0. \end{cases} \quad (3)$$

где

- $x \in \mathbb{Q}$ – входной параметр алгоритма;
- n – количество входных параметров торговой стратегии;
- $i = 1..n$ – индекс сходного параметра;
- $\text{step} \in \mathbb{Q}$ – точность входного параметра.

Система (3) является задачей поиска для генетического алгоритма с дискретной рекомбинацией. [3-А]

Особенность этого типа генетического алгоритма в том, что его вектора закодированы с помощью вещественных значений и к плюсам такого подхода можно отнести:

- 1 Задача проста для понимания.
- 2 ГА с дискретной рекомбинацией относительно легко реализовать.
- 3 Не требуются функции для дополнительного кодирования и декодирования векторов.
- 4 Отсутствуют дополнительные ограничения.

Однако, данный подход имеет и отрицательные стороны, которые, в основном сказываются на скорости работы алгоритма:

- 1 Увеличенный расход памяти, для хранения вещественных чисел.
- 2 Операции с вещественными числами выполняются дольше.
- 3 Потеря точности вычислений (особенности хранения вещественных чисел в памяти компьютера, при многократном использовании переменной происходит накопление ошибки в 16 знаке после запятой).

Так как скорость вычислений и точность вычислений являются приоритетными задачами в реализации ГА, будет предложен ряд преобразований для того, чтобы перевести задачу поиска из класса рациональных чисел (вещественный) в класс натуральных чисел (целочисленных).

После получения и применения ряда преобразований, задачу поиска вещественных входных параметров (3) было предложено свести к задаче поиска количества шагов от минимального кратного точности параметра значения, а в качестве функции декодирования использовать выражение (4):

$$\begin{cases} f(d(j_1), \dots, d(j_n)) \rightarrow \sup; \\ d(j_i) = \min_i + j_i \cdot \text{step}_i, \\ 0 \leq j_i \leq \left\lfloor \frac{\max_i}{\text{step}_i} \right\rfloor. \end{cases} \quad (4)$$

где

- $f(x)$ – целевая функция генетического алгоритма;
- $d(x)$ – функция декодирования гена ГА;
- $\min_i = \text{const} \in \mathbb{Q}$ – нижняя граница входного параметра;
- $\max_i = \text{const} \in \mathbb{Q}$ – верхняя граница входного параметра;
- $i \in \mathbb{Z}$ – индекс входного параметра;
- $j_i \in \mathbb{Z}$ – количество шагов точности;
- $\text{step}_i = \text{const} \in \mathbb{Q}$ – точность входного параметра.

ПРИМЕНЕНИЯ МЕТОДА ОСТРОВОВ ДЛЯ ГА

Для улучшения точности работы алгоритма было решено добавить логику островов. Основная идея заключается в том, что логика работы алгоритма разбивается на независимые острова, которые выполняются параллельную генерацию поколений. С определённой частотой, самые сильные особи каждого острова будут перемещены на другие острова с использованием карты миграции. Таким образом, решается ряд проблем:

- 1 Усиливается текущее поколение каждого острова после миграции.
- 2 Сохраняется разнообразие хромосом для расширения области поиска.
- 3 Данный подход является одним из вариантов защиты от появления альфа-особи на острове.
- 4 Добавляется возможность использования параллелизма в алгоритме.

Однако, данный подход будет иметь и ряд недостатков таких как:

- 1 Увеличивается общее время работы алгоритма.
- 2 Дополнительные вычислительные расходы на проведение миграции и сохранение миграционной карты.
- 3 Дополнительная нагрузка на процессор при параллельных вычислениях.
- 4 Возможность инфицирования островов мигрирующей альфа-особью.

Анализируя недостатки островной модели генетического алгоритма можно сказать, что дополнительные затраты на вычислительные расходы и общее время работы алгоритма будут компенсированы существенным улучшением области поиска после проведения операции миграции. Существенным недостатком островной модели является только пункт 4, благодаря которому есть вероятность «заразить» все острова модели и существенно ухудшить область поиска оптимального ответа.

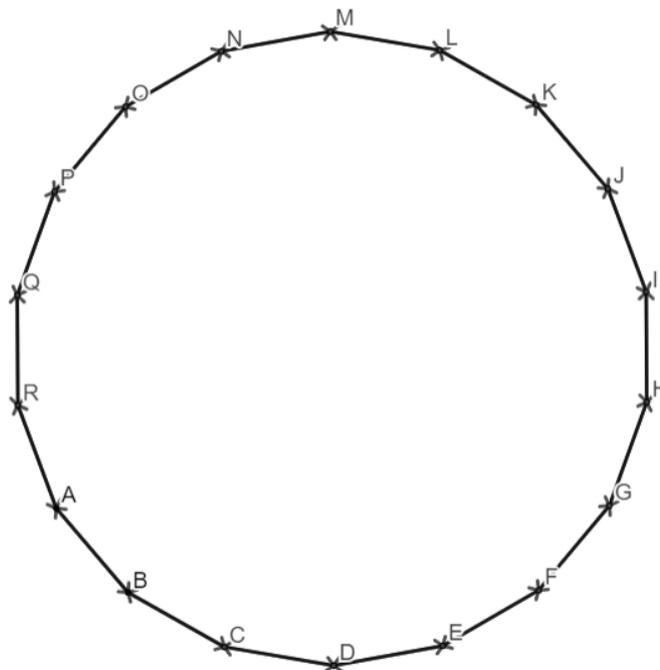


Рисунок 4 – Кольцевая двенаправленная миграционная карта для ГА

Для решения этой проблемы будет использоваться специальная миграционная карта, суть которой состоит в том, что в ней прописаны пути, по которым особи с одного острова должны мигрировать на другой. Если и в данном случае произойдёт сужение областей поиска всех островов к какой-то конкретной особи, то, возможно, эта особь является глобальным решением задачи.

Для уменьшения скорости распространения альфа-особи при её появлении на одном из островов, а также, для сохранения вариативности популяций между островами, была реализована модифицированная кольцевая миграционная карта. Её особенность состоит в том, что миграции между островами проходят не в одном заранее заданном направлении (например, по часовой стрелке), а сразу по всем направлениям. Так, каждый остров будет обмениваться своими лучшими особями с двумя соседями справа и слева от себя (рисунок 4).

Таблица 1 – Распределение различных операторов ГА между островами

Номер острова	Оператор отбора	Оператор выбора родительской пары	Оператор репродукции	Оператор мутации
A	Roulette	Panmixia	OneCrossover	Jump
B	Roulette	Panmixia	UniformCrossover	Jump
C	Roulette	Panmixia	AvrRecombination	Jump
D	Roulette	Inbreeding	OneCrossover	Jump
E	Roulette	Inbreeding	UniformCrossover	Jump
F	Roulette	Inbreeding	AvrRecombination	Jump
G	Roulette	Outbreeding	OneCrossover	Jump
H	Roulette	Outbreeding	UniformCrossover	Jump
I	Roulette	Outbreeding	AvrRecombination	Jump
J	Uniform	Panmixia	OneCrossover	Jump
K	Uniform	Panmixia	UniformCrossover	Jump
L	Uniform	Panmixia	AvrRecombination	Jump
M	Uniform	Inbreeding	OneCrossover	Jump
N	Uniform	Inbreeding	UniformCrossover	Jump
O	Uniform	Inbreeding	AvrRecombination	Jump
P	Uniform	Outbreeding	OneCrossover	Jump
Q	Uniform	Outbreeding	UniformCrossover	Jump
R	Uniform	Outbreeding	AvrRecombination	Jump

Итоговая версия генетического алгоритма содержит реализацию:

- двух операторов отбора (случайного - Uniform и пропорционального - Roulette);
- трёх операторов выбора родительской пары (инбридинг - Inbreeding, аутбридинг - Outbreeding и панмиксия - Panmixia);

- трёх операторов репродукции (одноточечного кроссовера - OneCrossover, единого кроссовера - UniformCrossover и кроссовера целой арифметической рекомбинации - AvrRecombination);
- одного оператора мутации (Jump мутация).

Итого существует 18 уникальных способов настройки ГА ($2 * 3 * 3 = 18$). Для того, чтобы все эти настройки использовались при работе генетического алгоритма, каждая полученная комбинация будет применена к определённому острову генетического алгоритма. Распределение настроек можно увидеть в таблице 1.1. Таким образом будет задействован полный спектр возможностей генетического алгоритма, и, возможно, один из островов найдёт самое лучшее решение задачи поиска оптимальной настройки торговой стратегии.

РЕЗУЛЬТАТЫ РАБОТЫ ГА С ОСТРОВАМИ

Результаты работы на уравнении с одной переменными

В качестве стартовой задачи найдём локальный максимум от функции с одной неизвестной переменной (5). Графическое представление функции представлено на рисунке 5.

$$y(x) = 5 - 24x + 17x^2 - \frac{11}{3}x^3 + \frac{1}{4}x^4, \quad (5)$$

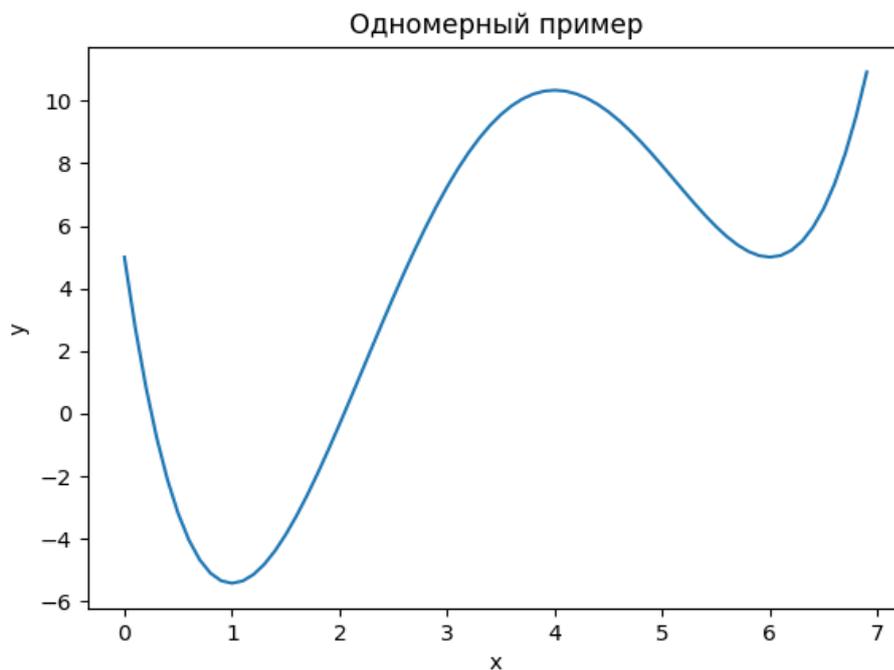


Рисунок 5 – График одномерной функции (5)

Переменную x задачи (5) будем искать на промежутке $[0, 7]$. Точность поиска будем увеличивать с каждой попытки и сравнивать результаты работы алгоритмов. Количество возможных вариантов решения задачи будет вычисляться по формуле (6):

$$TS = \prod_{i=1}^n \frac{\max_i - \min_i}{\text{step}_i}, \quad (6)$$

где

- TS – общее количество возможных вариантов решения задачи;
- i – индекс входного параметра поиска;
- n – общее количество входных параметров;
- \min_i – нижняя граница входного параметра;

- max_i – верхняя граница входного параметра;
- $step_i$ – точность входного параметра.

Таблица 2 – Результаты работы алгоритма полного перебора и ГА на задаче (5)

Step	TS	Время полного перебора, сек	Результат полного перебора	Время ГА	Результат ГА	Отклонение ГА (%)
1e-1	7*1e1	0.002517	5,4166667	0.010312	5.4166667	0.0 (0.00%)
1e-2	7*1e2	0.002518	5,4166667	0,070617	5,4166667	0.0 (0.00%)
1e-3	7*1e3	0.005638	5,4166667	0,541616	5,4166667	0.0 (0.00%)
1e-4	7*1e4	0,007196	5,4166667	0,538814	5,4166667	0.0 (0.00%)
1e-5	7*1e5	0,062130	5,4166667	0,530104	5,4166667	0.0 (0.00%)
1e-6	7*1e6	0,627611	5,4166667	0,524131	5,4166667	0.0 (0.00%)
1e-7	7*1e7	6,211779	5,41666667	0,544131	5,41666667	0.0 (0.00%)
1e-8	7*1e8	53,999341	5,416666667	0,566273	5,416666667	0.0 (0.00%)
1e-9	7*1e9	591,186594	5,4166666667	0,555604	5,4166666667	0.0 (0.00%)

Результаты работы на уравнении с двумя переменными

Следующий тест будет выполняться на функции с двумя переменными (7). В данном наборе тестов будут проверяться взаимодействие многогенных хромосом, работа с отрицательными и положительными числами и поведение генетического алгоритма при наличии нескольких локальных экстремумов (рисунок 6).

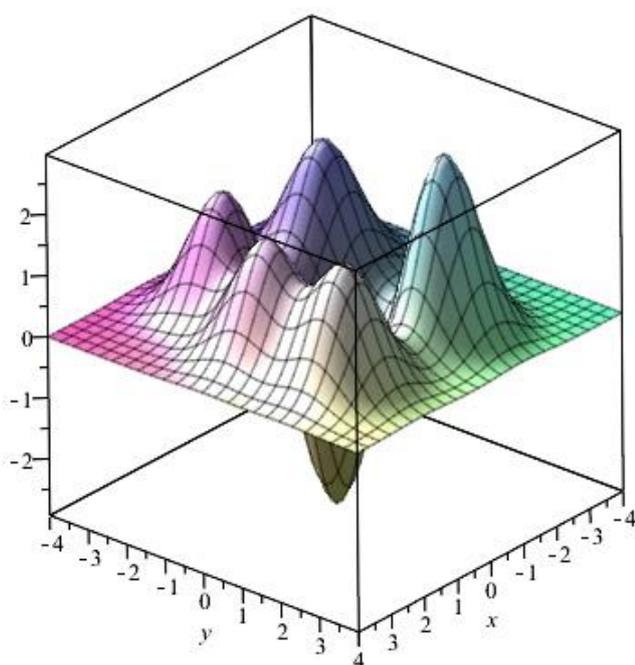


Рисунок 6 – График двумерной функции (7)

$$z = F(x, y, 0, 0, -3) + F(x, y, 2, 2, 2) + F(x, y, -2, -2, 2) + F(x, y, -1, 2, 3) + F(x, y, 1, -2, 2) + F(x, y, 2, 0, 2), \quad (7)$$

где

$$F(x, y, a, b, c) = c \cdot e^{-(x-a)^2 - (y-b)^2}, \quad (8)$$

Переменные x и y для задачи (7) будем искать в промежутке от -3 до 3. Результаты работы алгоритмов представлены в таблице 3.

Таблица 3 – Результаты работы алгоритма полного перебора и ГА на задаче (7)

Step	TS	Время полного перебора, сек	Результат полного перебора	Время ГА	Результат ГА	Отклонения ГА (%)
1e-1	3364	0,003158	2,98003759	0,010859	2,98003759	0.0 (0.00%)
1e-2	357604	0,039900	2,98060376	0,074959	2,98060376	0.0 (0.00%)
1e-3	35976004	3,697948	2,98067053	0,680951	2,98067053	0.0 (0.00%)
1e-4	3599760004	332,451188	2,98067080	0,629279	2,98067080	0.0 (0.00%)

Результаты работы на исторических данных

В завершении проведём тестирование ГА, интегрированного в программные продукты семейства TickTrader на реальных данных. Для этого возьмём исторические котировки по символу EURUSD с периодом в 1 минуту за промежуток с 12.01.2021 по 05.01.2022 (1). Начальный баланс аккаунта 10000\$. В качестве функции приспособленности будет использована торговая стратегия МА. Поиск осуществляется на параметрах $MovingPeriod=[1..100]$ и $MovingShift=[1..100]$.

Таблица 4 – Результаты работы алгоритма полного перебора и ГА на исторических данных символа EURUSD

Step	TS	Время полного перебора, сек	Результат полного перебора	Время ГА	Результат ГА	Отклонения ГА (%)
20	25	100,02245	10011.342	186,61863	10011.342	0.0 (0.00%)
10	100	400.92518	10011.342	189.12313	10011.342	0.0 (0.00%)
5	400	1603.99881	10011.342	194.09483	10011.342	0.0 (0.00%)
2	2500	5000,07177	10011.342	188.00953	10011.342	0.0 (0.00%)
1	10000	12021,00930	10011.342	200.23211	10011.342	0.0 (0.00%)

Как видно из таблицы 4 ГА смог найти оптимальный ответ. Но, т.к. оптимальный ответ всего один и общее количество возможных комбинаций в максимальном случае всего 10000, данные результаты тестирования не так представительны, как могли бы быть. Однако в текущем случае невозможно было увеличить область поиска, т.к. финальное время работы алгоритма полного перебора на 10000 комбинациях даже с добавлением операций асинхронного и многопоточного программирования уже составляло более 3 часов реального времени.

ЗАКЛЮЧЕНИЕ

Главным результатом работы данной диссертации стала разработка генетического алгоритма, адаптированного под задачу поиска оптимальных настроек торговых стратегий. Алгоритм был спроектирован таким образом, чтобы его настройка и запуск требовали минимальных знаний и действий от конечного пользователя, при этом, алгоритм старается адаптироваться к поставленной задаче поиска с сохранением баланса между средним временем работы и точностью полученного решения. После разработки и тестирования данный программный продукт был интегрирован в семейство программных продуктов TickTrader компании SoftFX. Также, в процессе написания диссертации, соискатель проанализировал доменные предметные области, изучил продукты конкурентных компаний, отзывы и отчёты об ошибках пользователей продукта TickTrader AlgoStudio, провёл ряд исследований, измерений и тестов. Для создания оптимальной версии генетического алгоритма соискателям была адаптирована и оптимизирована стандартная математическая модель генетического алгоритма с дискретной рекомбинацией. Во время исследований и проведения экспериментов было опубликовано три работы в сборниках трудов и материалов международных конференций по теме диссертации.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1-А Кузьма, Ю. В. Алгоритмы и методы машинного обучения для оценки устойчивости физически неклонированных функций к криптографическим атакам / Кузьма Ю. В., Хлопцев А. А. // Информационные технологии и системы 2020 (ИТС 2020) = Information Technologies and Systems 2020 (ITS 2020) : материалы международной научной конференции, Минск, 18 ноября 2020 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2020. – С. 177–178.

2-А Хлопцев, А. А. Генетический алгоритм для оптимизации подбора параметров торговых ботов / Хлопцев А. А., Кузьма Ю. В. // Информационные технологии и системы 2020 (ИТС 2020) = Information Technologies and Systems 2020 (ITS 2020) : материалы международной научной конференции, Минск, 18 ноября 2020 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2020. – С. 88–89.

3-А Хлопцев, А. А. Сравнение эффективности различных операторов выбора родительской пары в генетическом алгоритме с дискретной рекомбинацией / А. А. Хлопцев // Электронные системы и технологии [Электронный ресурс] : сборник материалов 58-й научной конференции аспирантов, магистрантов и студентов БГУИР, Минск, 18-22 апреля 2022 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Д. В. Лихаческий [и др.]. – Минск, 2022. – С. 39-42. – Режим доступа : <https://libeldoc.bsuir.by/handle/123456789/46926>.