

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(Н И У « Б е л Г У »)**

**ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ**

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ
ПОИСКА ОПТИМАЛЬНОЙ ТОРГОВОЙ СТРАТЕГИИ ДЛЯ РЫНКА
ЦЕННЫХ БУМАГ**

Магистерская диссертация
обучающегося по направлению подготовки
09.04.02 Информационные системы и технологии
очной формы обучения
группы 07001635
Гахова Богдана Романовича

Научный руководитель
к.т.н., доцент
Титов А.И.

Рецензент
к.т.н., доцент
Путивцева Н.П.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 Исследование процесса разработки оптимальной торговой стратегии для рынка ценных бумаг	8
1.1 Исследование процесса разработки торговой стратегии	8
1.2 Оценка эффективности различных торговых стратегий.....	14
1.2 Актуальность разрабатываемой модели	19
1.4 Общая характеристика организации решения задачи.....	21
2 Моделирование и проектирование информационной системы	24
2.1 Описание исследуемых стратегий.....	24
2.2 Описание генетического алгоритма.....	28
2.3 Разработка модели поиска оптимальной торговой стратегии средствами генетического алгоритма.....	32
2.3 Требования к разрабатываемой информационной системе.....	36
2.4 Выбор архитектуры ИС	37
2.5 Выбор программных средств реализации	38
2.6 Проектирование автоматизированной системы поиска оптимальной стратегии для рынка ценных бумаг.....	40
2 Описание разработанной автоматизированной системы поиска оптимальной торговой стратегии для рынка ценных бумаг.....	45
3.1 Описание разработанного пользовательского интерфейса	45
3.2 Поиск оптимальных параметров торговой стратегии с помощью разработанной информационной системы	54
ЗАКЛЮЧЕНИЕ	59
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	61
ПРИЛОЖЕНИЕ	64

РЕФЕРАТ

Разработка автоматизированной системы поиска оптимальной торговой стратегии для рынка ценных бумаг – Гахов Б.Р., магистерская диссертация, Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 63, включает приложение, количество рисунков 35, количество таблиц 1, количество использованных источников 30.

КЛЮЧЕВЫЕ СЛОВА: ценные бумаги, алгоритмическая торговая система, оптимизация процесса поиска, рынок ценных бумаг, моделирование, генетический алгоритм, проектирование информационной системы, разработка информационной системы.

ОБЪЕКТ ИССЛЕДОВАНИЯ: процесс поиска оптимальной торговой стратегии для рынка ценных бумаг.

ПРЕДМЕТ ИССЛЕДОВАНИЯ: модель поиска оптимальной торговой стратегии для рынка ценных бумаг.

ЦЕЛЬ РАБОТЫ: оптимизация средствами генетического алгоритма поиска торговой стратегии для рынка ценных бумаг.

ЗАДАЧИ ИССЛЕДОВАНИЯ: уменьшение вычислительных мощностей, требуемых для оптимизации параметров торговой стратегии; автоматизация процесса поиска оптимальной торговой стратегии для рынка ценных бумаг.

МЕТОДЫ ИССЛЕДОВАНИЯ: системный анализ; анализ литературы, математическое и имитационное моделирование, теория вероятности и математической статистики.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: в процессе выполнения магистерской диссертации было проведено общее исследование процесса разработки торговой стратегии и описаны принципы оценки их эффективности. Проведен анализ существующих аналогичных информационных систем и выявлены их недостатки. Обоснована необходимость в разработке новой модели поиска оптимальной торговой стратегии. Проведена общая характеристика организации решения задачи.

Описаны виды исследуемых торговых стратегий. Были рассмотрены основные этапы и понятия классического генетического алгоритма. На основании проведенного анализа была разработана модель поиска оптимальной торговой стратегии средствами генетического алгоритма для сокращения количества исследуемых вариантов. Для реализации разработанной модели было принято решение о необходимости разработки

новой информационной системы с обоснованием требований, выбором архитектуры приложения и средства реализации. Выполнено проектирование автоматизированной системы с применением методологии DFD, с разработкой контекстной диаграммы и выполнением декомпозиции. Создана диаграмма классов в соответствии с нотацией UML и разработана схема базы данных для хранения сведений о инструментах, отчетах, котировках, секторах.

Представлено описание графического интерфейса пользователя разработанной информационной системы с подробным описание функционала используемых элементов. Программное приложение позволяет строить графики оптимизации, а также выполнять поиск наилучших параметров стратегий с предоставлением подробной информации о исследуемых торговых стратегиях и смоделированных сделках. Проведено сравнение количества исследуемых наборов параметров стратегий с использованием разработанной модели, основанной на генетическом алгоритме, и методом полного перебора.

По результатам исследования и проектирования была реализована автоматизированная система поиска оптимальной торговой стратегии для рынка ценных бумаг. Для разработки системы использовалась современная среда разработки Microsoft Visual Studio 2017, что позволило быстро создать пользовательский интерфейс, вести удобную и качественную разработку, без применения дополнительных решений.

ВВЕДЕНИЕ

В настоящее время существует множество способов для увеличения денежных средств, одним из которых является ведение финансовой деятельности на рынке ценных бумаг. Множество компаний предлагают свои услуги по сохранению и приумножению капитала клиента, используя различные финансовые инструменты. Трейдер, человек извлекающий доход из движения цен на различные активы, и различные крупные участники торгов, в каждый момент выбирают стратегии для ведения собственной торговой деятельности на рынке ценных бумаг, количество вариантов этих стратегий огромно, и у большинства из них есть некоторые параметры, которые существенно влияют на показатели эффективности работы этой стратегии.

При выборе собственной торговой стратегии, трейдеру необходимо в первую очередь изучить финансовые инструменты, используя которые он хочет получить прибыль. В зависимости от динамики цен, характерные именно для этих инструментов, необходимо выработать набор правил для принятия решения о использовании актива. Стоит обратить внимание на уже существующие торговые стратегии для выбранных или схожих инструментов, изучить их параметры и влияние, которые они оказывают на результат работы стратегии. На сегодняшний день существует множество систем, позволяющих реализовать работу торговой стратегии с использованием специализированных программных и программно-аппаратных систем, чтобы предоставить трейдеру наилучшие условия для торговой деятельности и избавить от рутинных вычислений, необходимых для определения моментов для осуществления торговли. Используя специализированное программное обеспечение, имеется возможность реализовать множество стратегий, с различными параметрами, но моделирование работы стратегии и поиск оптимального набора стратегий с их параметрами требует значительных

вычислительных мощностей, а при большом количестве исследуемых вариантов стратегий оптимизация может длиться неприемлемо долгий временной период даже на самых производительных системах. Актуальность темы обусловлена необходимостью оптимизации процесса поиска оптимальной торговой стратегии для рынка ценных бумаг [1].

Объект исследования: процесс поиска оптимальной торговой стратегии для рынка ценных бумаг.

Предметом исследования магистерской работы является модель поиска оптимальной торговой стратегии для рынка ценных бумаг.

Целью магистерской диссертации является оптимизация поиска торговой стратегии для рынка ценных бумаг средствами генетического алгоритма.

Для выполнения поставленной задачи необходимо создание информационной системы поиска оптимальной торговой стратегии для рынка ценных бумаг, которая должна облегчить специалистам поиск оптимального набора параметров торговой стратегии, обеспечить инструментами для оптимизации разработанных стратегий и предоставить возможность изменения используемых стратегий и доработку методов оптимизации.

Научная новизна состоит в разработке оригинальной модели поиска оптимальной торговой стратегии для рынка ценных бумаг.

В ходе выполнения магистерской диссертации необходимо решить следующие задачи:

- провести общее исследование предметной области;
- определить направление автоматизации поиска оптимальной торговой стратегии;
- разработать модель поиска оптимальной торговой стратегии для рынка ценных бумаг;
- осуществить проектирование и реализацию информационной системы;

– выполнить описание поиска оптимальных параметров торговой стратегии с помощью разработанной информационной системы.

Выпускная квалификационная работа включает введение, три раздела, заключение, список использованных источников, приложение.

В первом разделе рассматривается существующее состояние предметной области: описываются характеристики и особенности деятельности трейдера, виды торговых стратегий, риски, связанные с торговой деятельностью и методы оценки разработанной торговой стратегии, анализируются недостатки существующих информационных систем для поиска оптимальной торговой стратегии, и обосновывается предложение по устранению найденных недостатков и, внедрению новых подходов.

Второй раздел содержит описание используемых торговых стратегий, описание работы классического генетического алгоритма и его особенности применения для поиска оптимальной торговой стратегии для рынка ценных бумаг. Выполнено обоснование проектных решений по информационному, программному и технологическому обеспечению задачи, а также проектирование информационной системы поиска оптимальной стратегии.

Третий раздел содержит описание разработанной автоматизированной системы в том числе описание разработанного пользовательского интерфейса. Выполнен поиск оптимальной торговой стратегии для рынка ценных бумаг.

В заключении подведены итоги решения задач, сделаны выводы, отражена практическая значимость достигнутых результатов.

Магистерская диссертация написана на 63 листах, содержит 35 рисунков, 1 таблицу и приложение.

1 Исследование процесса разработки оптимальной торговой стратегии для рынка ценных бумаг

1.1 Исследование процесса разработки торговой стратегии

Приняв решение о необходимости использования финансовых инструментов для извлечения прибыли, трейдер должен выбрать брокера или биржу для осуществления торговой деятельности. Часто брокер или биржа дают возможность использовать множество различных финансовых инструментов, так как чем больше людей использует их услуги, тем больше прибыль, получаемая с комиссий за совершение торговых операций.

Биржей называется юридическое лицо, обеспечивающее регулярное функционирование организованного рынка биржевых товаров, валют, ценных бумаг и производных финансовых инструментов. Биржа предоставляет свои услуги не для всех клиентов, частыми ограничениями являются работа исключительно с юридическим лицом, значительный объем каждой сделки и проторгованный объем за месяц или другой срок. Эти ограничения созданы, чтобы исполнение наиболее крупных и важных сделок проходили быстрее, и клиенты с наибольшими активами могли получить наиболее выгодные условия для своей финансовой деятельности. Для ведения деятельности на рынке ценных бумаг с меньшими активами, чем у наиболее крупных банков и правительствами государств существуют брокеры – это юридические или физические лица, выполняющие роль посредника между продавцом и покупателем. Брокер ведет свою деятельность с группой клиентов, и накапливает сделки заказы от клиентов, на покупку или продажу доступных ему активов. При достижении некоего порогового значения, он совершает сделки, на необходимых инструментах у доступных ему бирж или других брокеров и клиентам параметры сделки, взывая комиссию за свои услуги или получая постоянный доход, в зависимости от договора с клиентом. Таким

образом может выстраиваться цепочка сделок, от трейдера до биржи и чем звеньев больше, тем с менее выгодными условиями будет исполнена заявка трейдера [1].

Получив доступ к выбранным торговым инструментам у брокера или непосредственно у биржи, человек занимающийся трейдингом должен ознакомиться с условиями использования интересующих его финансовых инструментов. К наиболее важным следует отнести такие параметры как:

- информация о торговых сессиях, которая представляет собой временные интервалы, во время которых доступны торговые операции. Данные интервалы могут прерываться в течении дня, а также могут быть перерывы на выходные дни;

- комиссии, которые взимает брокер или биржа. Комиссии могут браться с каждой сделки, или при вводе и/или выводе средств со счета и другими способами;

- минимальный и максимальный объем, шаг объема – данные показатели означают какую сумму можно выставить при открытии ордеров, а также какому числу должна быть кратна указанная сумма;

- доступ к операциям, этот показатель означает какие типы ордеров доступны для трейдера.

К основным типам ордеров относятся покупка (англ. Buy или Long длинная позиция) и продажа (англ. Sell или Short короткая позиция) торгового инструмента. Покупка означает приобретение актива, с ожиданием роста его цены и извлечения прибыли. Продажа означает продажу актива и извлечение прибыли при падении цены. Следует отметить, что продажа актива которого нет в наличии у трейдера предполагает взятие его в долг у брокера или биржи – если такая услуга предоставляется брокером или биржей, то это будет отмечено в договоре или в спецификации инструмента [1].

Часто брокер или биржа предоставляет возможность использования отложенных ордеров различного типа. Отложенный ордер – это распоряжение

брокерской компании купить или продать финансовый инструмент в будущем по заданным условиям. Существует следующие типы отложенных ордеров:

– Buy Limit – торговый приказ на покупку по цене "Ask" равной или меньшей, чем указанная в ордере. При этом текущий уровень цен больше значения установленного в ордере. Обычно ордера этого типа выставляются в расчете на то, что цена инструмента, опустившись до определенного уровня, начнет расти;

– Buy Stop – торговый приказ на покупку по цене "Ask" равной или большей, чем указанная в ордере. При этом текущий уровень цен меньше значения установленного в ордере. Обычно ордера этого типа выставляются в расчете на то, что цена инструмента преодолет некий уровень и продолжит свой рост;

– Sell Limit – торговый приказ на продажу по цене "Bid" равной или большей, чем указанная в ордере. При этом текущий уровень цен меньше значения установленного в ордере. Обычно ордера этого типа выставляются в расчете на то, что цена инструмента, поднявшись до определенного уровня, начнет снижаться;

– Sell Stop – торговый приказ на продажу по цене "Bid" равной или меньшей, чем указанная в ордере. При этом текущий уровень цен больше значения установленного ордера. Обычно ордера этого типа выставляются в расчете на то, что цена инструмента достигнет определенного уровня и продолжит снижаться.

– Buy Stop Limit – этот вид ордера сочетает в себе первые два типа, являясь стоп ордером на установку лимитного ордера на покупку ("Buy Limit"). Как только будущая цена "Ask" достигнет стоп-уровня, указанного в этом ордере (поле "Цена"), будет выставлен "Buy Limit" приказ на уровне, указанном в поле "Цена Stop Limit". Стоп-уровень выставляется выше текущей цены Ask, а цена Stop Limit - ниже стоп-уровня.

– Sell Stop Limit – этот вид ордера является стоп ордером на установку лимитного ордера на продажу ("Sell Limit"). Как только будущая цена "Bid" достигнет стоп-уровня, указанного в этом ордере (поле "Цена"), будет выставлен "Sell Limit" приказ на уровне, указанным в поле "Цена Stop Limit". Стоп-уровень выставляется ниже текущей цены Bid, а цена Stop Limit - выше стоп-уровня [2].

На рисунке 1.1 представлен график, на котором точкой слева обозначена текущая цена, подписанной точкой обозначена цены открытия различных типов ордеров, а стрелкой указано направление цены для получения прибыли по указанному ордеру.



Рисунок 1.1 – Типы ордеров

При открытии ордера на выбранном финансовом инструменте указывается необходимая цена и лот (объем) сделки, также могут быть другие дополнительные параметры. При успешном выполнении ордера брокер или биржа сообщает, что сделка прошла успешно и позиция трейдера изменена. Позицией называется текущий состав открытых трейдером сделок на текущий момент.

Торговая стратегия представляет собой набор правил для выполнения торговых операций для выбранного финансового инструмента. Для полного понимания стратегии необходимо в каждый момент времени решить, какие финансовые операции необходимы для соответствия торговой стратегии. К ним могут относиться следующие действия:

- открытие или закрытие ордеров указанного объема;
- изменение открытых ордеров, почти всегда это действие может быть реализовано через открытие или закрытие ордеров противоположного типа.

В текущее время крупные биржи и брокеры предоставляют интерфейс для удаленного выполнения торговых операций. Состав компонентов торговой платформы MetaTrader 4 представлен на рисунке 1.2.

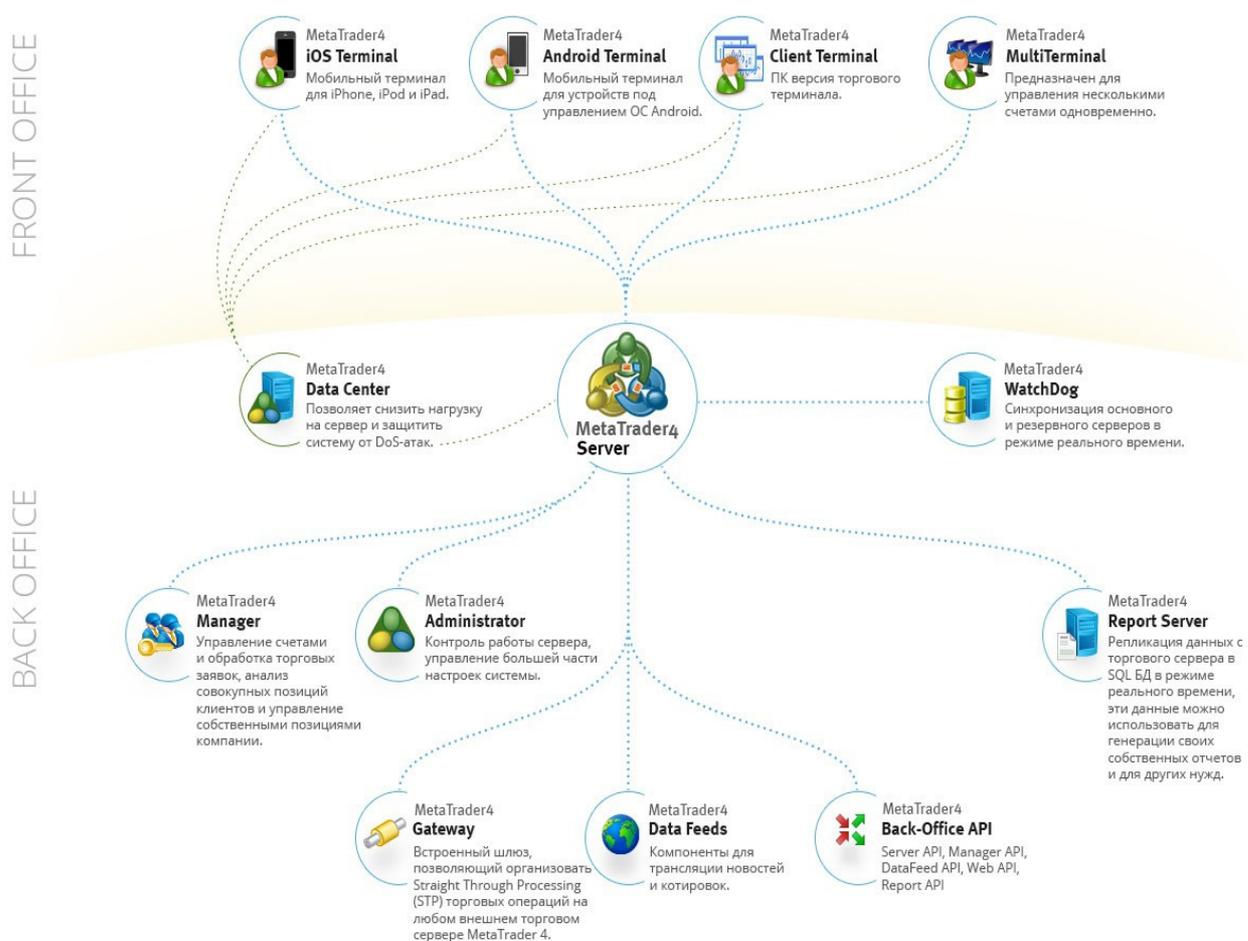


Рисунок 1.2 – Состав компонентов торговой платформы MetaTrader 4

Также большинство предоставляют набор API (интерфейс прикладного программирования) и/или программное обеспечение, позволяющее автоматизировать часть или всю деятельность трейдера.

Для эффективной работы на финансовых рынках и избавления трейдера от рутинной работы необходимо разработать успешную торговую систему. Постоянно придерживаться выработанной торговой системы в ручном режиме очень сложно, из-за значительной роли эмоционального фактора. Механические торговые системы лишены этого недостатка [3].

Торговые информационные системы способны не только выдавать рекомендательные торговые сигналы, но и полностью управлять торговой деятельностью в реальном режиме времени.

Торговый советник позволяет:

- автоматизировать аналитические и торговые процессы;
- оперативно проводить технический анализ ценовых данных и на основе полученных сигналов управлять торговой деятельностью;
- проведению технического анализа и выполнению всех торговых операций;
- совершать аналитические и торговые операции по любым финансовым инструментам и периодам независимо от того, открыт соответствующий график или нет;
- контролировать риски и потери.

Таким образом, автоматизация деятельности трейдера устраняет проблемы лишних трудозатрат, экономит время ручной торговли, формирует точные и математически обоснованные сигналы [3].

1.2 Оценка эффективности различных торговых стратегий

Существует множество видов торговых стратегий, все они основаны на определенных принципах, областях экономики, математики, психологии и любых других науках, которые применяют разные трейдеры.

Наиболее широкое распространение получили следующие виды торговых стратегий:

- трендовая – одна из распространенных, популярных и наиболее прибыльных видов торговых систем на различных финансовых инструментах, которая строится на предположениях о тенденции роста или падения, если она имеет место, прослеживается ее устойчивость и продолжение (тренд, от которого и берет начало название система, рассматривают относительно определенного временного периода;

- флетовая – строится на предположении, что цены двигаются в рамках коридора или канала, имеющего ограничения уровнями сопротивления сверху и поддержки снизу, поэтому около уровня поддержки необходимо покупать и продавая вблизи уровня сопротивления;

- контртрендовые стратегии, которая основана на определении точки разворота тренда – правильно определенный момент разворота позволяет получить максимальную прибыль при обратном движении, но применение этих стратегий сопряжено с большим количеством ложных входов (этот вид стратегий требует дополнительных навыков и опыта трейдера);

- стратегии с применением свечных паттернов (формаций) – системы, основанные на паттернах, которые являются характерным сочетанием ценовой модели и ее производной за промежуток времени, имеющей предсказуемое продолжение по статистическим данным;

- стратегии, использующие информацию из новостей, финансовых отчетов и других важных источников информации, существенно влияющих на цену финансового инструмента. В зависимости от идеи, целью этих стратегий

может служить совершение торговых операций как до получения необходимой информации, так и после, в ожидание определенной финансовой деятельности различных участников рынка ценных бумаг [1].

Ключевым аспектом деятельности каждого трейдера является знание принципов торговли на интересующих его финансовых инструментах. Для того чтобы извлекать прибыль из движения цены финансового инструмента необходимо иметь доступ к торговле им на бирже или у брокера и знать множество статистических и фундаментальных данных о нем. К таковым могут относиться следующие показатели:

- природа финансового инструмента – сведения о том, почему этот инструмент интересен различным людям;
- средняя цена за различные временные интервалы;
- проторгованный объем, характеризующий общее количество актива, который был куплен на рынке;
- различные математические показатели, такие как дисперсия цены за временной интервал, значение корреляции с другими финансовыми инструментами и т.д.
- различные фундаментальные данные, касающиеся новостей об активе, сведения о заинтересованности в нем различных крупных инвесторов, крупных компаний или других лиц способные сильно повлиять на его будущую цену [4].

Наиболее полезной и значимой является классификация всех активов на базовые и вторичные – также называемые деривативами. Производный финансовый инструмент, дериватив– договор, по которому стороны получают право или берут обязательство выполнить некоторые действия в отношении базового актива. К базовым активам относятся:

- ценные бумаги;
- товары;
- валюта;

- процентные ставки;
- уровень инфляции;
- официальная статистическая информация
- другие.

В отличие от прямого договора купли/продажи, дериватив формален и стандартизирован, изначально предусматривает возможность минимум для одной из сторон свободно продавать данный контракт, то есть является одним из вариантов ценных бумаг. Цена дериватива и характер её изменения обычно тесно связаны с ценой базового актива, но не обязательно совпадают.

Дериватив имеет следующие характеристики:

- его стоимость меняется вслед за изменением цены базового актива (процентной ставки, цены товара или ценной бумаги, обменного курса, индекса цен или ставок, кредитного рейтинга или кредитного индекса, другой переменной);
- для его приобретения достаточно небольших первоначальных затрат по сравнению с другими инструментами, цены на которые аналогичным образом реагируют на изменения рыночной конъюнктуры;
- расчёты по нему осуществляются в будущем.

В зависимости от всех характеристик финансового актива часть движения цены может быть заранее известной или прогнозируемой, что всегда необходимо учитывать при торговле на бирже.

У каждого финансового инструмента существует множество особенностей из-за уникальных спецификаций и возможностей, которые они дают трейдеру или инвестору. В связи с этим затруднительное или невозможно точное моделирование поведения цены или моделирование работы торговых стратегий. Поэтому торговая деятельность сопровождается множеством рисков, которые должен учитывать трейдер [2].

Вся экономическая деятельность связана с риском. Риск – это вероятность наступления какого-либо события. Риск связан с

неопределенностью результатов той или иной деятельности или бездеятельности. Все они находятся в определенных отношениях друг с другом, между ними устанавливаются связи и зависимости. Предприятие, например, зависит от поставщиков и от потребителей своей продукции, от банка, через которые осуществляются расчеты, от государства, которое устанавливает «правила игры», то есть для него всегда существует элемент неопределенности в связи с деятельностью контрагентов, а также и в связи с воздействием других факторов: природных, социальных, политических и т.д.

Далее представлены наиболее значимые на сегодняшний день типы финансовых рисков.

Финансовые риски, связанные с операциями с ценными бумагами – это вероятность наступления потерь в связи с высокой степенью неопределенности результатов таких операций, а также влияния на них множества экономических и неэкономических факторов, в том числе случайных.

Потери в результате реализации финансовых рисков в операциях с ценными бумагами могут проявиться в разных формах, а именно:

- прямые финансовые потери (например, ценная бумага, купленная по высокой цене, не может быть реализована без убытков в случае падения ее курсовой стоимости);
- упущенная выгода (например, инвестор вместо того, чтобы купить акцию, которая продемонстрировала в будущем высокий рост курсовой стоимости, оставил денежные средства на расчетном счете в банке);
- снижение доходности (например, в случае приобретения облигации с «плавающим» купонным доходом в ситуации, когда происходит понижение рыночных процентных ставок).

Макроэкономические риски, которые возникают в результате факторов, действующих на уровне экономики в целом, однако разные ценные бумаги по-разному восприимчивы к действию этих факторов [5].

Кредитный риск – это вероятность наступления потерь из-за того, что эмитент не сможет выполнить свои обязательства по выплате основной суммы долга, оформленного ценными бумагами, или процентов по нему. Кредитный риск относится только к тем ценным бумагам, по которым эмитент имеет фиксированные обязательства.

Риски управления портфелем и технические риски – это вероятность наступления потерь из-за неправильных действий управляющего инвестиционным портфелем, а из-за сбоев в системе поставки ценных бумаг, в системе расчетов по сделкам с ценными бумагами, а также сбоями в операционной системе, начиная от ошибок операциониста и заканчивая компьютерным мошенничеством. Потери от этих рисков могут быть очень велики, причем в настоящее время, когда использование компьютерной техники и Интернета позволяет очень быстро передавать информацию до огромного количества пользователей, значимость технических ошибок возрастает на много порядков.

Системный (рыночный) риск – это вероятность наступления потерь в результате падения рынка ценных бумаг в целом. Этот вид риска не связан с какой-то конкретной ценной бумагой, он является общим для всех вложений в ценные бумаги. Инвестор не может каким-либо образом повлиять на значение этого риска, снизить его, поэтому системный риск является недиверсифицируемым и непонижаемым.

Так же стоит выделить несистемный, или специфический риск – это вероятность наступления потерь в связи с операциями с конкретной ценной бумагой (портфелем ценных бумаг), это обобщенное понятие, характеризующее все виды рисков, связанных с конкретными ценными бумагами. Этот риск не связан непосредственно с состоянием рынка, напротив, он связан с деятельностью, положением на рынке, качеством менеджмента компании, но наряду с системным риском, свойственным всему рынку в целом, является частью общего риска инвестиций в ценные бумаги.

Диверсификация вложений – это покупка ценных бумаг разных эмитентов, непосредственно не связанных между собой. Несистемный риск является диверсифицируемым и понижаемым. Это означает, что инвестор, производя отбор ценных бумаг, удовлетворяющих его с точки зрения риска, диверсифицируя свои вложения, может понизить значение данного риска.

Таким образом диверсификация является базовым понятием уменьшения рисков, при правильном использовании которой инвестор, трейдер, инвестиционная компания и другие могут сильно понизить вероятность резких финансовых потерь, связанных со своей деятельностью [4].

Оценка эффективности инвестиционного портфеля – составляющая инвестиционного процесса, заключающаяся в периодическом анализе функционирования инвестиционного портфеля в терминах доходности и риска.

С точки зрения снижения рисков, наилучшим вложением денежных средств является приобретение государственных облигаций, обеспечивающих безрисковую процентную ставку, однако отсутствие риска сказывается на уровне доходности, редко покрывающем убытки, связанные с инфляционными процессами. Несмотря на это, безрисковая процентная ставка является эталоном для оценки эффективности любого типа инвестиционных стратегий [6].

1.2 Актуальность разрабатываемой модели

Каждая торговая стратегия имеет набор параметров и правил для совершения торговых операций в зависимости от значения цены или нескольких цен в определенный моменты времени. Задача полного перебора значений параметров стратегий, а также определение объема депозита для ведения финансовой деятельности на рынке – весьма ресурсоемка,

оптимизация может длиться годами даже на самой производительной аппаратуре.

Наиболее устоявшимся и привычным способом оптимизации является максимальное сокращение перебираемых значений параметров, с использованием экспертных знаний. После этого этапа выполняется полный или частичный перебор по некоторым правилам, чтобы получить результат за приемлемое время. Недостатком данного метода является сильная зависимость результата от опыта и знаний эксперта или экспертов, которые определяют целесообразные для тестирования диапазоны значений параметров и их точность. Так же следует учесть, что при увеличении числа параметров рост числа вариантов происходит в геометрической прогрессии, и эффективность данного метода снижается [1].

Следует учесть, что при неверном понимании целей и методов оптимизации повышается риск, что после составления новой “оптимальной” стратегии при её использовании не будут достигнуты ожидаемые показатели, или даже результат будет убыточным для трейдера. Это связано с тем, что при оптимизации часть убыточных сделок, которые могут быть обнаружены только после их совершения были убраны из результатов оптимальной стратегии различными методами. Это может быть прекращение торговли в определенные моменты времени или при достижении некоторых значений цен, фиксирование убытков в неоправданных местах, выбивающихся из общей логики стратегии и многие другие методы, прям или косвенным способом отсеивающие негативные сделки для повышения значений целевого параметра.

Для моделирования процесса оптимизации значений параметров, необходимо выполнить следующие действия:

– реализовать функцию (или функции), целью которой является оценка эффективности торговой стратегии по итогу работы каждой торговой стратегии, значение которых будет максимизировать в ходе выполнения оптимизации;

- написать исходный код исследуемых торговых стратегий и отметить параметры, оптимизация значений которых приведет к улучшению значений выбранного целевого показателя;
- для каждого оптимизируемого значения параметра необходимо указать область допустимых значений и точность;
- выполнить генерацию всех сочетаний значений параметров с исходным диапазоном;
- выполнить тестирование стратегий с указанными значениями параметров;
- выполнить ранжирование вариантов по наибольшему значению финансового показателя и вывести их в качестве результатов работы [3].

Задача поиска оптимальной торговой стратегии возникает при наличии множества эффективных торговых стратегий и необходимости сформировать наиболее качественную результирующую систему для открытия ордеров на выбранных финансовых инструментах. Если выполнять полный перебор значений параметров, то процесс оптимизации может длиться весьма долго независимо от используемой вычислительной мощности, что неприемлемо для трейдера в большинстве случаев. Следовательно, задача сокращения количества просчитываемых вариантов алгоритмом является актуальной.

1.4 Общая характеристика организации решения задачи

На сегодняшний день, широкое применение получила информационная система компании MetaQuotes. Возможность использования оптимизации значений торговых стратегий встроена в торговый терминал MetaTrader версии 4 и 5 компании MetaQuotes. Программист разрабатывающий автоматизированную торговую стратегию с использованием этого инструмента использует язык mql, данный основывается на языке с++, большая часть средств языка доступна, также имеются дополнительные типы и функции для автоматизации процесса торговли. Для выполнения

оптимизации, в исходном коде есть внешние переменные, значения которые требуют оптимизации необходимо обозначить ключевым словом `extern` или `input` при объявлении, после чего необходимо скомпилировать исходный код и перейти в терминал MetaTrader, выбрать интересующий торговый инструмент, временной период для оптимизации и обозначить параметры для оптимизации, их стартовые, конечные значения и шаг приращения для выполнения перебора, после чего будет выполнен полный перебор всех сочетаний параметров и максимизация целевого параметра. Функцию вычисления данного параметра можно выбрать из стандартных или написать в исходном коде стратегии. Также существует возможность генетического поиска, для сокращения количества сочетаний параметров до фиксированного количества исследуемых вариантов, но максимальное количество исследуемых наборов параметров ограничено константным значением, а код используемого генетического алгоритма недоступен для просмотра и изменения [3].

Недостатков при использовании данного инструмента множество, наиболее существенными является невозможность изменения финансового инструмента в MetaTrader 5 и использование только одного источника для котировок используемых при создании ордеров. Это означает возможность использовать инструменты, к которым выполнено подключение в текущий момент и предоставлены только одним брокером. Даже если имеются другие данные для оптимизации или есть несколько аккаунтов у брокеров, предоставляющие различные финансовые инструменты – эти данные можно только считать, но провести по ним сделки невозможно стандартными средствами встроенного языка `mql`. В MetaTrader 4 невозможен запуск оптимизации или тестирование стратегий открывающие ордера на нескольких торговых инструментах. Также имеются некоторые ограничения при написании исходного кода советника на языках `mql4` и `mql5`.

Таким образом не имеется средств для оптимизации параметров торговой стратегии, с открытым исходным кодом, гибкими настройками метода перебора параметров.

Для решения указанной проблемы предлагается использовать генетический алгоритм, позволяющий сократить количество исследуемых вариантов наборов параметров. Так как исходные коды к системам MetaTrader не предоставлены и систем с подобным функционалом нет в открытом виде, целесообразно разработать собственную систему для тестирования и подбора оптимальных параметров торговой стратегии для рынка ценных бумаг.

Выводы по первому разделу: в данном разделе были исследованы особенности разработки торговой стратегии и описаны принципы оценки их эффективности. Проведен анализ существующих аналогичных информационных систем и выявлены их недостатки. Обоснована необходимость в разработке новой модели поиска оптимальной торговой стратегии. Проведена общая характеристика организации решения задачи.

2 Моделирование и проектирование информационной системы

2.1 Описание исследуемых стратегий

Торговая стратегия - это совокупность правил, приемов и инструментов анализа, которыми руководствуется трейдер при заключении сделок. Торговые стратегии помогают получить максимальную прибыль и минимизировать убытки во время торговли. Это обязательный и незаменимый элемент финансового рынка при работе с валютной парой или конкретной акцией на рынке. На данный момент торговые стратегии не создаются, все биржевые рынки давно уже изучены х.

Торговля на основе технических индикаторов широко распространены при проведении торговых операций на рынке валют и акций. В моделировании были использованы следующие базовые стратегии:

- торговля на основе индикатора MACD;
- торговля на основе индикатора Stochastic Oscillator;
- торговля на основе индикатора Relative Strength Index.

Индикатор MACD позволяет увидеть согласованность краткосрочного и долгосрочного консенсуса масс по поводу текущей цены, который, в свою очередь, может быть различным. Стоит отметить, что у MACD гораздо меньше ложных сигналов, чем у торговой системы из одного скользящего среднего.

Если линия MACD пересекает сигнальную линию MACD снизу вверх, то совершается покупка. Также нелишним будет опциональное выставление стоп-заявки за последним локальным минимумом. При пересечении линии MACD сигнальной линии MACD сверху вниз совершается продажа (если трейдер находился в длинной позиции, то она закрывается и открывается шорт). Стоп опционально выставляется выше последнего локального максимума [1].

Торговые сигналы индикатора изображены на рисунке 2.1.



Рисунок 2.1 – Торговые сигналы индикатора MACD

На приведенном рисунке время для открытия позиции на продажу обозначены стрелкой вниз, позиции для покупки – стрелкой вверх. По методике торговли на основе MACD данный индикатор можно отнести к реверсным. Это означает, что при выполнении методологии индикатора трейдер, закрывая одну позицию, открывает другую – противоположную первой.

Технический индикатор Moving Average Convergence/Divergence определяется путем вычитания экспоненциального скользящего среднего с большим периодом из скользящего среднего меньшего периода. Затем на график MACD пунктиром наносится его простое скользящее среднее, которое выполняет роль сигнальной линии.

Другим популярным техническим индикатором является стохастический осциллятор, сопоставляющий цену закрытия с диапазоном цен за выбранный период времени.

Этот осциллятор даёт сигналы на покупку в момент, когда линии индикатора выходят из зоны перепроданности, в идеальном варианте – когда быстрая линия пересекает медленную снизу вверх. В этой ситуации стоит совершать покупку на открытии свечи после выхода стохастический осциллятор из перепроданности, размещая стоп за последним локальным

минимумом. Сигналы на продажу индикатор подает при выходе из зоны перекупленности, в идеальном случае – когда быстрая линия индикатора пересекает медленную сверху вниз. В этой ситуации стоит совершать продажу на открытии свечи после выхода осциллятора из перекупленности, размещая стоп за последним локальным максимумом [3].

На рисунке 2.2 представлены совершения сделок на выходе из зон перекупленности и перепроданности по стохастическому осциллятору по стохастическому осциллятору.



Рисунок 2.2 – Сигналы совершения по стохастическому осциллятору

На приведенном рисунке время для открытия позиции на продажу обозначены стрелкой вниз, позиции для покупки – стрелкой вверх.

Для расчета стохастического осциллятора используются четыре переменные:

- периоды $%K$ — число единичных периодов, используемых для расчета стохастического осциллятора;

- периоды замедления $%K$ — величина, определяющая степень внутренней сглаженности линии $%K$, причем значение 1 дает быстрый стохастический осциллятор, а значение 3 - медленный;

– периоды $%D$ — число единичных периодов, используемых для расчета скользящего среднего линии $%K$;

– метод $%D$ — метод сглаживания (экспоненциальный, простой, сглаженный или взвешенный), используемый при расчете $%D$.

Значение $%K$ рассчитывается следующим образом:

$$\%K = (CLOSE - MIN (LOW (\%K))) / (MAX (HIGH (\%K)) - MIN (LOW (\%K))) * 100,$$

где $CLOSE$ - сегодняшняя цена закрытия;

$MIN (LOW (\%K))$ - наименьший минимум за число периодов $%K$;

$MAX (HIGH (\%K))$ - наибольший максимум за число периодов $%K$.

Скользящее среднее $%D$ рассчитывается как простая скользящая средняя.

Технический индикатор Индекс Относительной Силы (Relative Strength Index, RSI) это следующий за ценой осциллятор, который колеблется в диапазоне от 0 до 100 (рисунок 2.3).



Рисунок 2.3 - Технический индикатор Индекс Относительной Силы

Один из распространенных методов анализа индикатора Relative Strength Index состоит в поиске расхождений, при которых цена образует

новый максимум, а RSI не удается преодолеть уровень своего предыдущего максимума. Подобное расхождение свидетельствует о вероятности разворота цен. Если затем индикатор поворачивает вниз и опускается ниже своей впадины, то он завершает так называемый "неудавшийся размах" (failure swing). Этот неудавшийся размах считается подтверждением скорого разворота цен. Основная формула расчета технического индикатора Relative Strength Index:

$$RSI = 100 - (100 / (1 + U / D)),$$

где U — среднее значение положительных ценовых изменений;

D — среднее значение отрицательных ценовых изменений [6].

Таким образом, были рассмотрены несколько базовых стратегий для проведения оптимизации значений их параметров в дальнейшем.

2.2 Описание генетического алгоритма

При описании генетических алгоритмов используются определения, заимствованные из генетики. Например, речь идет о популяции особей, а в качестве базовых понятий применяются ген, хромосома, генотип, фенотип, аллель. Также используются соответствующие этим терминам определения из технического лексикона, в частности, цепь, двоичная последовательность, структура.

Популяция - это конечное множество особей.

Особи, входящие в популяцию, в генетических алгоритмах представляются хромосомами с закодированным в них множествами параметров задачи, т.е. решений, которые иначе называются точками в пространстве поиска (search points). В некоторых работах особи называются организмами.

Хромосомы, или цепочки, или кодовые последовательности - это упорядоченные последовательности генов.

Ген (также называемый свойством, знаком или детектором) - это атомарный элемент генотипа, в частности, хромосомы.

Генотип или структура - это набор хромосом данной особи. Следовательно, особями популяции могут быть генотипы либо единичные хромосомы (в довольно распространенном случае, когда генотип состоит из одной хромосомы).

Фенотип - это набор значений, соответствующих данному генотипу, т.е. декодированная структура или множество параметров задачи (решение, точка пространства поиска) [7].

Основной (классический) генетический алгоритм (также называемый элементарным или простым генетическим алгоритмом) состоит из следующих шагов:

- инициализация, или выбор исходной популяции хромосом;
- оценка приспособленности хромосом в популяции;
- проверка условия останова алгоритма;
- селекция хромосом;
- применение генетических операторов;
- формирование новой популяции;
- выбор «наилучшей» хромосомы.

В генетических алгоритмах за передачу признаков новым особям отвечает оператор скрещивания (который так же называется кроссовер или кроссинговер), который действует следующим образом:

- из популяции выбираются две особи, которые будут родителями;
- случайным образом определяется точка разрыва;
- потомок определяется как объединение первого и второго родителя .

Схема работы генетического оператора представлена на рисунке 2.4:

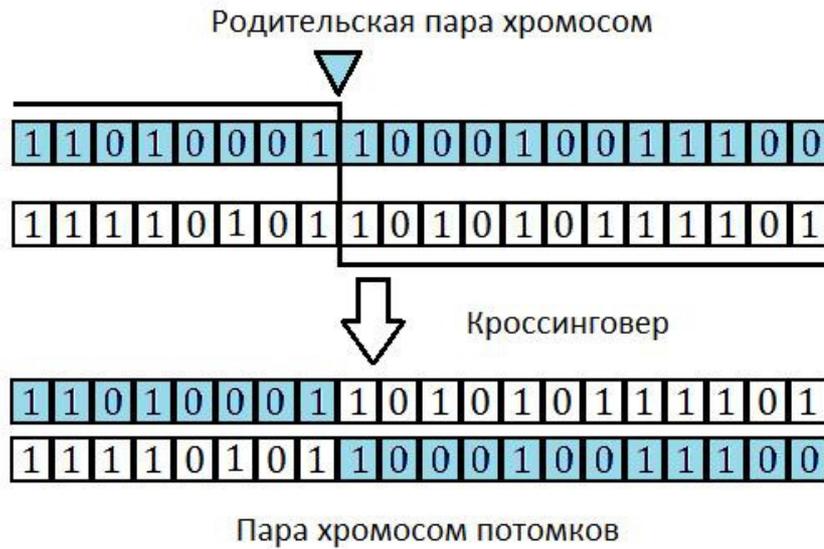


Рисунок 2.4 – Схема работы кроссинговера

Так как в результате получается две последовательности, то с вероятностью 0.5 одна из результирующих хромосом выбирается в качестве потомка [8].

Для поддержания разнообразия особей в популяции также используется оператор мутации. При его использовании каждый бит в хромосоме инвертируется с определенной вероятностью.

Схема работы оператора мутации представлена на рисунке 2.5.

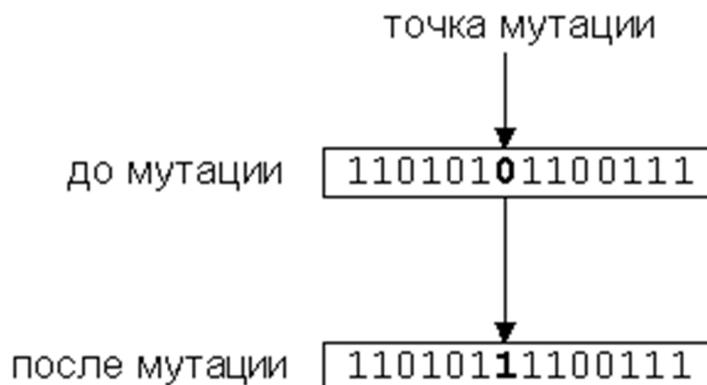


Рисунок 2.5 – Схема работы мутации

Также может быть применен оператор инверсии с определенной вероятностью, при его выполнении хромосома делится на две части и затем они меняются местами. Схема работы оператора инверсии представлена на рисунке 2.6.

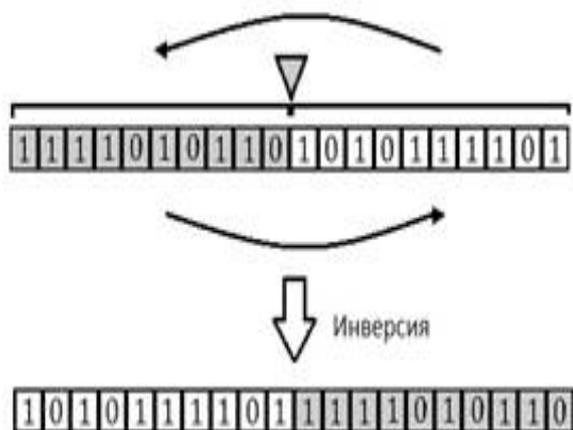


Рисунок 2.6 – Схема работы оператора инверсии

Важную роль в успешной работе алгоритма играет этап отбора родительских особей. Наиболее часто используется метод, который называется рулеткой. При использовании этого алгоритма, вероятность выбора хромосома в качестве родителя определяется как отношение приспособленности хромосомы, к приспособленности популяции. Использование этого метода приводит к тому, что вероятность передать признаки более приспособленным особям возрастает. Другой часто используемый метод – турнирный отбор. При его использовании случайно выбирается несколько особей, победителем выбирается особь с большей приспособленностью [7].

Таким образом, в популяции присутствует несколько групп особей, внутри которых возможно эффективное применение операторов. Критерием для прекращения работы алгоритма, может быть выбрано достижение необходимой точности значения функции приспособленности популяции, что

означает, что изменение значения фитнес функции меньше некоторого заданного числа.

2.3 Разработка модели поиска оптимальной торговой стратегии средствами генетического алгоритма

Чтобы обеспечить выполнение генетического алгоритма для поиска оптимальной стратегии необходимо выполнить предварительные этапы:

- определить метод представления стратегии в виде хромосомы;
- определить генетические операторы;
- определить условие завершения работы алгоритма;
- выбор списка стратегий.

На этапе написания торговой стратегии необходимо отметить те параметры, которые могут быть использованы при выполнении оптимизации. Наиболее частыми типами данными, которые необходимо оптимизировать, являются целые числа, числа с плавающей точкой и строки.

Для кодирования числовых типов, перед началом оптимизации, необходимо определиться с их областью возможных значений, а также точностью представленной в виде шага. Значение шага S означает, что значения параметра в промежутке $[p-s/2; p+s/2)$, может интерпретироваться как p (где p некоторое допустимое число из области допустимых значений), так как указанная точность гарантирует, что эти значения отличаются незначительно [9].

После определения области допустимых значений и шага, необходимо представить каждый параметр в виде последовательности битов. На первом этапе необходимо определить количество значений исследуемого параметра, и посчитать количество необходимых бит для кодирования параметра. Для этого можно использовать формулу:

$$N = \text{Ceil}(\log_2(C)); \quad (1)$$

где C – количество возможных значений,

N – количество бит необходимых для кодирования параметра,

Ceil – округление для ближайшего большего числа.

После чего значение параметра можно представить в двоичном виде, как

$$P_{\text{binary}} = \text{Bin}(P_{\text{decimal}} - P_{\text{startValuedecimal}}); \quad (2)$$

где P_{decimal} – значение параметра в десятичном виде;

$P_{\text{startValuedecimal}}$ – минимальное значение параметра в десятичном виде;

Bin – функция для кодирования десятичного числа в последовательность бит;

P_{binary} – битовая последовательность числа.

Таким образом, чтобы записать хромосому необходимо представить все параметры в виде последовательности.

После определения функции для получения хромосом, необходимо определиться со способом, которым родительские признаки передаются потомкам.

Входными данными для работы генетического алгоритма является набор стратегий и их параметры с возможными ограничениями, а также количество особей в популяции k , и вероятности для выполнения генетических операторов. На каждом этапе генетического алгоритма генерируется новая популяция, высчитывается приспособленность каждой особи в популяции и всей популяции, после чего проверяется условие завершения алгоритма. Если оно выполняется, то алгоритм вернет наиболее приспособленную популяцию в качестве результата работы.

Модель поиска оптимальной торговой стратегии представлена в виде алгоритма на рисунке 2.7.

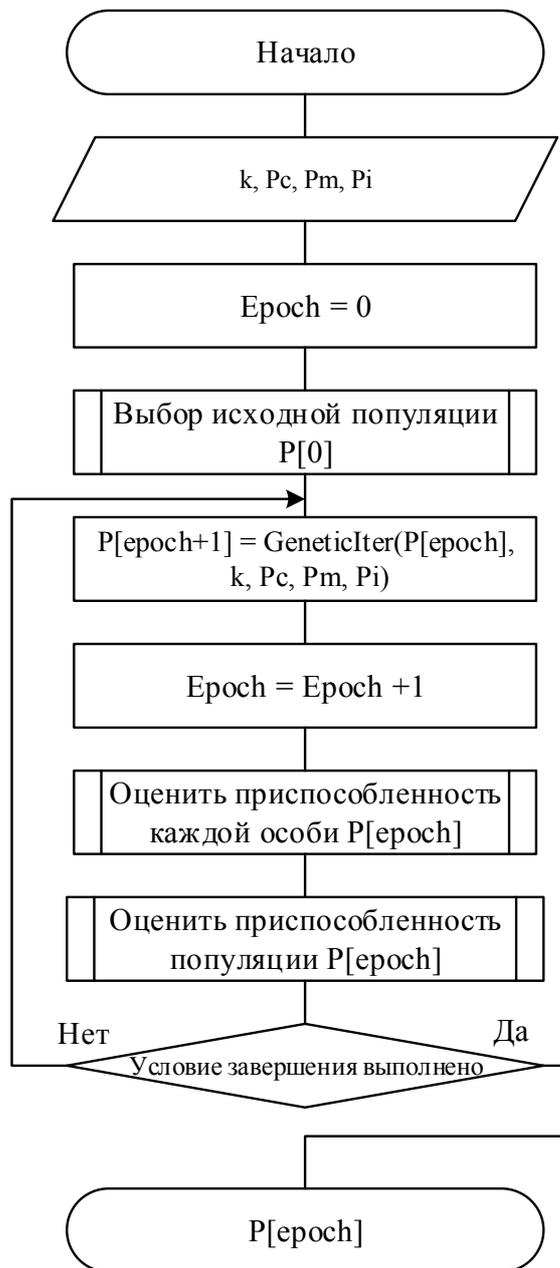


Рисунок 2.7 - Схема генетического алгоритма

В начале работы алгоритма необходимо инициализировать стартовую популяцию $P[0]$ из k случайных особей, а также инициализировать начальный момент $epoch=0$. После инициализации необходимо вычислить функции приспособленности для каждой особи в $P[epoch]$ и всей популяции в целом. Это значение описывает насколько хорошо подходит особь, описываемая данной хромосомой для решения задачи. Если условие завершения не выполнено, то необходимо получить новую популяцию и провести новый этап генетического алгоритма.

Схема алгоритма получения новой популяции представлен на рисунке

2.8.

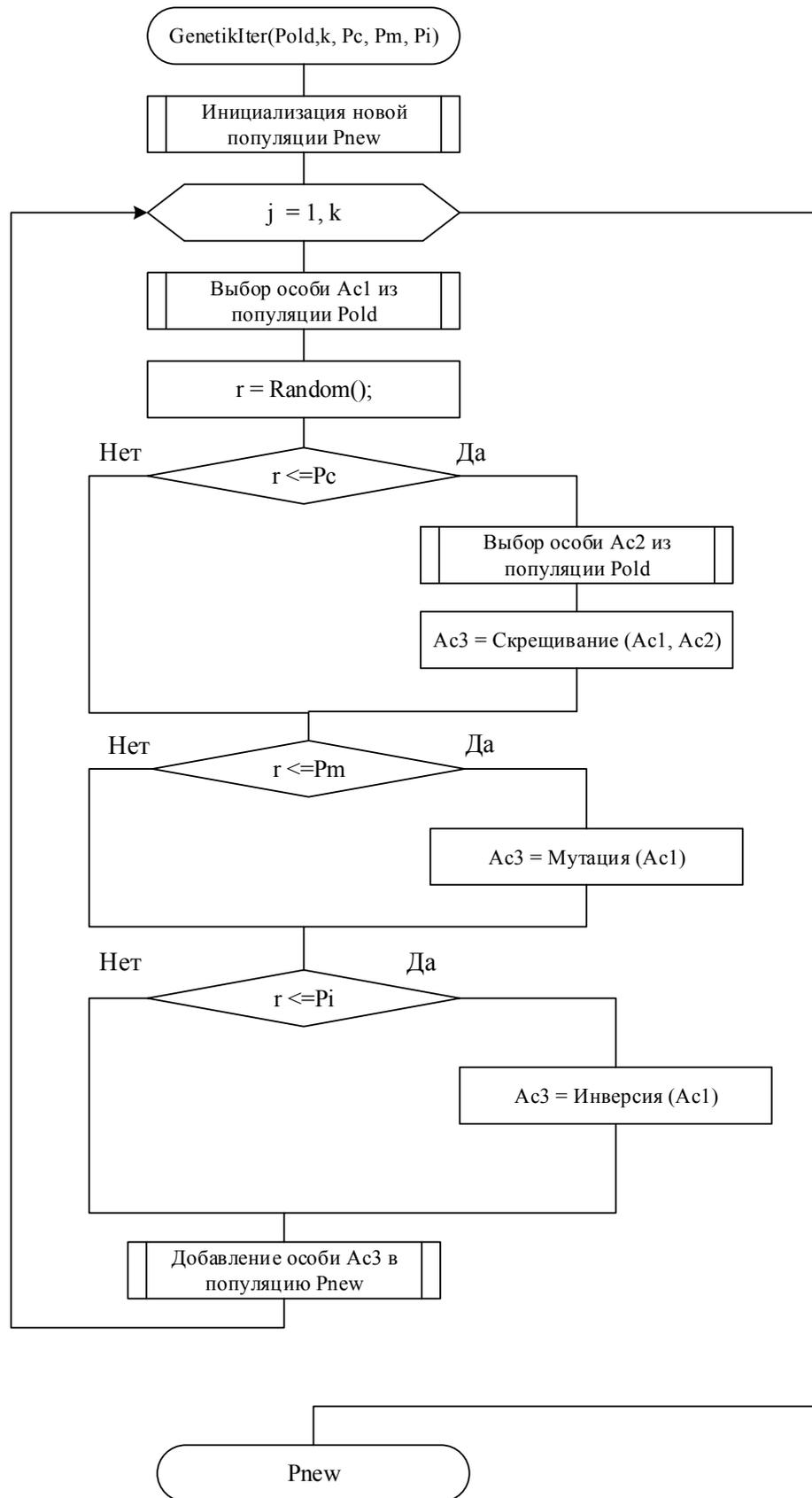


Рисунок 2.8 – Блок-схема алгоритма получения новой популяции

Входными данными для функции является предыдущая популяция P_{old} , число особей в популяции и вероятности для выполнения кроссовера P_c , мутации P_m , инверсии P_i .

Перед началом получения новых особей, необходимо инициализировать пустую популяцию P_{new} , в которую будут помещены новые особи. После этого, на каждом этапе цикла необходимо выбрать особь из популяции, и в соответствии с вероятностью P_c выполнить оператор кроссовера, предварительно выбрав еще одну особь. Если скрещивание не было произведено, то выполнить с вероятностью P_m мутацию, и если не была произведена мутация – выполнить инверсию с вероятностью P_i , если не было выполнено ни одно из условий, то включить выбранную особь без изменений в новую популяцию.

Цикл продолжается, пока не будет получено k особей в популяции и в качестве результата функция возвращает новую популяцию P_{new} .

2.3 Требования к разрабатываемой информационной системе

Данная система подразумевает решение следующих задач: хранение сведений о торговых стратегиях и исторических данных о финансовых инструментах, а также использование этой информации для составления наиболее эффективной стратегии торговли по выбранному финансовому показателю или группе показателей.

Таким образом, при разработке системы необходимо решить ряд задач, связанных с хранением большего объема данных, а также оперативного доступа к ним. В настоящее время не существует готовых программных решений.

Назначение разработки – автоматизированная система поиска оптимальной торговой стратегии для рынка ценных бумаг предназначена для решения следующих задач:

- хранения исторических сведений о котировках по различным торговым инструментам;

- возможность написания торговых стратегий различной сложности с различным числом оптимизируемых параметров;

- возможность написания собственного критерия для оценки эффективности торговой стратегии или использование одного из стандартных.

Требования к функциональным характеристикам – система должна обеспечивать следующие функции:

- добавление и удаление котировок из БД;

- оптимизация торговых стратегий;

- формирование отчетов по торговой стратегии;

- возможность сохранения отчетов о оптимизации торговых стратегий.

Входной информацией системы является информация о торговых инструментах, исторические сведения цен, а также торговые стратегии, их параметры, и их диапазоны значений.

Выходной информацией системы является отчет об итогах оптимизации, список параметров и стратегий для оптимальной финансовой деятельности по заданным критериям и исходным данным.

Требования к надежности – система должна:

- проводить валидацию вводимой информации;

- предупреждать о некорректных действиях пользователя при работе с системой;

- обеспечивать целостность данных.

2.4 Выбор архитектуры ИС

Разработка автоматизированной торговой стратегии часто ведется небольшой группой программистов. Для быстрой проверки базовой идеи, необходимо многократно использовать уже имеющийся код, написанный в

команде. Данные о котировках, хранятся локально у каждого разработчика для быстрого использования при тестировании, но существуют различные системы для синхронизации данных. Все эти данные можно хранить на сервере, при этом каждый будет синхронизировать данные перед началом тестирования. Для избегания повторного выполнения тестирования, необходимо хранить результирующие сделки в общей базе данных, используя в качестве ключей информацию, которая необходима для однозначного определения отчета тестирования стратегии. Чтобы результаты тестирования торговой стратегии, были получены в максимально короткие сроки программное обеспечение должно обладать высокой производительностью, таким образом, наиболее подходящей архитектурой информационной системы является нативное десктопное клиентское приложение, с возможностью подключения к СУБД в локальной сети [10].

При разработке сложной системы со множеством хранимых состояний, которые необходимы для выполнения генетического алгоритма, хранения в памяти множества значений параметров к каждой стратегии, обеспечение работы целесообразно использовать парадигму объектно-ориентированного программирования, что позволит разделить сложную систему на множество связанных компонент, взаимодействующих с пользователем системы и между собой.

2.5 Выбор программных средств реализации

AllFusion Process Modeler 7 (ранее BPwin) – инструмент для моделирования, анализа, документирования и оптимизации бизнес-процессов. AllFusion Process Modeler 7 можно использовать для графического представления бизнес-процессов. Графически представленная схема выполнения работ, обмена информацией, документооборота визуализирует модель бизнес-процесса. Графическое изложение этой информации позволяет

перевести задачи управления организацией из области сложного ремесла в сферу инженерных технологий [11].

AllFusion ERwin Data Modeler (ранее ERwin) – CASE-средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных и хранилища. Модели данных помогают визуализировать структуру данных, обеспечивая эффективный процесс организации, управления и администрирования таких аспектов деятельности предприятия, как уровень сложности данных, технологий баз данных и среды развертывания [12].

Для проектирования диаграммы классов был выбран Microsoft Visio – векторный графический редактор, редактор диаграмм и блок-схем для Windows. Данный редактор предоставляет возможность создавать диаграммы различных схем, в том числе нотации UML.

После того как трейдинг стал доступен широкой аудитории, основная часть пользовательского интерфейса была создана для операционных систем семейства Windows компании Microsoft, так как на конец 90х годов являлись наиболее распространёнными система с продуманным пользовательским интерфейсом. Такая тенденция сохранилась к нынешнему времени, и большинство трейдеров использующие выполняющие сделки самостоятельно и разработчики алгоритмов торговли используют операционные системы семейства Windows. Компания Microsoft в настоящий момент продвигает несколько языков программирования для быстрой и эффективной разработки для их операционной системы, наиболее полной справкой и эффективной производительностью итогового исполняемой программы, является язык C#. Для разработки на этом языке наиболее популярной средой интегрированной разработки является «Microsoft Visual Studio», в наиболее актуальных версиях этой среды имеется множество средств улучшения качества и скорости разработки приложений на языке C#, таких как удобный текстовый редактор с авто дополнением используемых лексем (функция «IntelliSense»),

инструменты для отладки кода, графический “конструктор” для создания оконных приложений и многое другое [13].

Выбранные средства позволяют выполнить необходимые этапы для проектирования автоматизированной системы поиска оптимальной торговой стратегии для рынка ценных бумаг, а также разработки необходимых модулей и графического интерфейса пользователя.

2.6 Проектирование автоматизированной системы поиска оптимальной стратегии для рынка ценных бумаг

Для проектирования системы была выбрана нотация DFD – диаграммы потоков данных. Контекстная диаграмма системы представлена на рисунке 2.9.

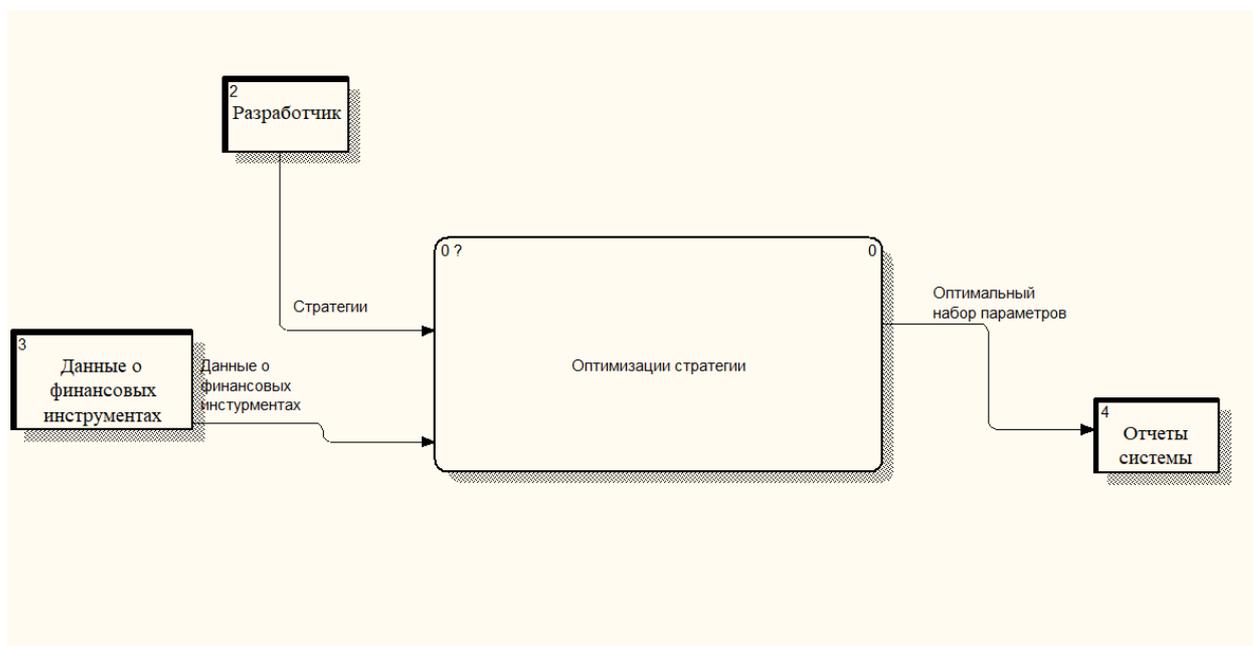


Рисунок 2.9 – Контекстная диаграмма оптимизации стратегии

Таким образом, необходимо получить в результате работы системы оптимальные значения параметров.

Для того, чтобы оптимизировать значения параметров, необходимо получить результаты тестирования, представляющие собой выбранный

финансовый показатель для оценки эффективности торговой стратегии. После чего необходимо поменять значения используемых параметров и пересчитать эти показатели для всех стратегий. Чтобы значения выбранного финансового показателя в результате было приемлемым, необходимо или рассчитать финансовые значения при всех существующих наборах параметров, или использовать генетический алгоритм [8].

На рисунке 2.10 представлена детализация процесса оптимизации стратегии.

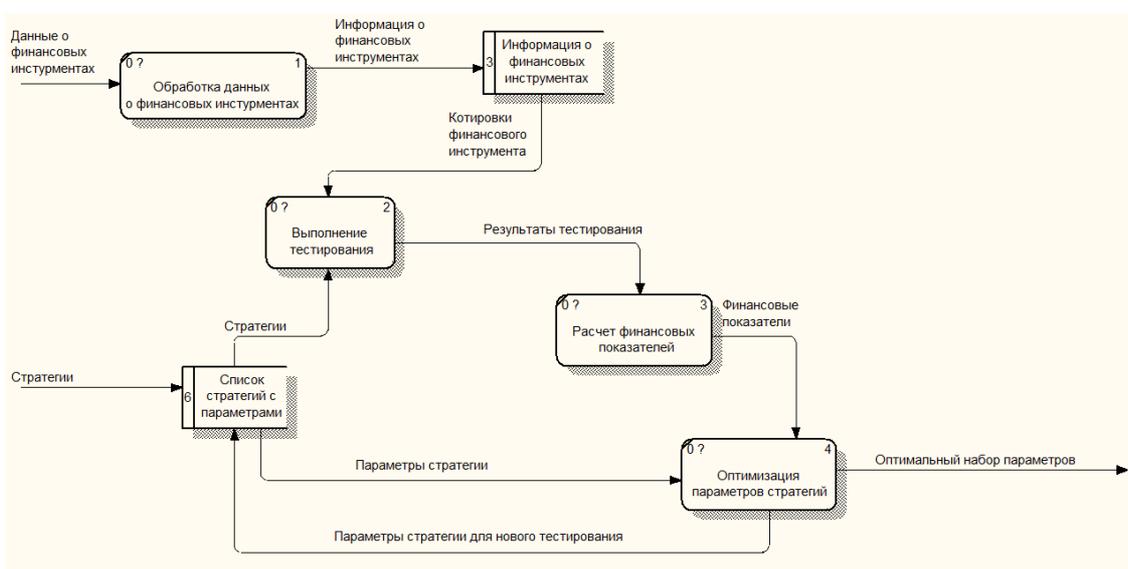


Рисунок 2.10 – Диаграмма-декомпозиция оптимизации стратегии

Важно, что значения оптимизированного показателя с каждой итерацией должны приближаться к некоторому локальному максимуму, иначе процесс будет бесконечен. На этот случай необходимо вывести сообщение разработчику о том, что процесс бесконечен и прервать работу системы [7].

Для решения задачи в соответствии с представленными диаграммами, необходимо выполнить декомпозицию системы. Исходя из поставленных целей и созданных диаграмм, необходимо создать несколько классов и абстрактный класс для решения задачи.

На рисунке 2.11 представлена диаграмма классов в соответствии с нотацией UML, в которой приведены основные классы системы, их атрибуты и методы, важные для обеспечения взаимодействия.

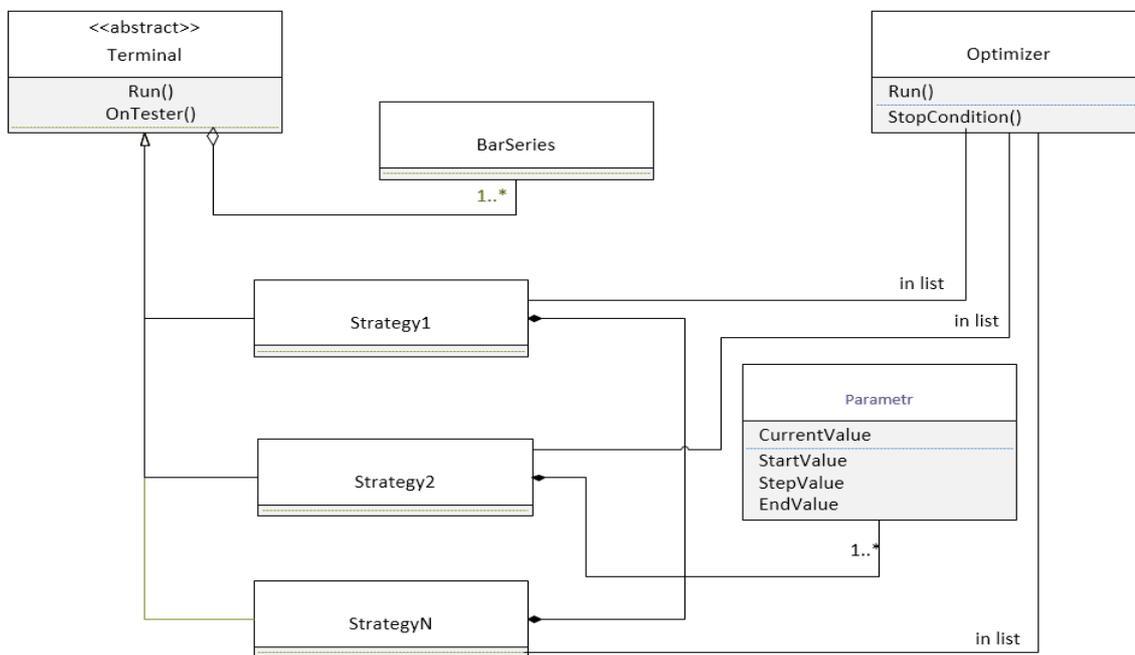


Рисунок 2.11 – Диаграмма классов системы оптимизации торговой стратегии

Базовым абстрактным классом системы будет являться Terminal, для наследования данного класса, необходимо реализовать методы OnTick и OnTester. Первый метод вызывается в методе Run (который реализован в родительском классе), при каждом новом вызове аргументом метода будет являться новая котировка. Таким образом абстрактный класс Terminal подразумевает добавление в список своих параметров и набора исторических сведений о котировках одного или нескольких финансовых инструментов. Котировки представляют собой сведения о цене финансового инструмента, которые хранятся в виде упорядоченной по времени последовательности японских свечей, со значениями проторгованного объема и OHLC (цены открытия, максимума, минимума и закрытия на соответствующем периоде). Второй метод OnTester() – необходим для оценки эффективности торговой

стратегии. Данный метод должен учитывать закрытые ордера по окончании тестирования и на их основе вычисляться, стандартно, в качестве оценки торговой стратегии – используется прибыльность, которую при необходимости разработчик может переопределить при создании отдельной стратегии. Результатом данного метода является действительное число, возвращаемое методом Run по завершению тестирования [14].

Таким образом, каждая стратегия может быть представлена в виде класса, который наследует абстрактный класс Terminal, после выполнения метода Run будет получен список сделок по которому станет возможным подсчет финансового показателя.

Для оптимизации стратегии важно, чтобы результат зависел от параметров, которыми являются экземпляры класса Parameter. Их использование необходимо для того, чтобы иметь возможность оптимизировать стратегию независимо от внутренней реализации метода OnTick() и вызываемых внутри методов.

После создания группы стратегий и добавления экземпляров стратегий в класс Optimizer, в данном классе вызывается метод Run(), в котором вызывается метод Run() каждой стратегии, после тестирования всех стратегий, можно посчитать значение фитнес функции для популяции стратегий, и принять решение о возможности остановки работы алгоритма.

Для хранения исторических сведений о котировках необходимо создать базу данных с ценами всех финансовых инструментов.

В данной базе необходимо хранить сведения о инструменте, отчетах, котировках, сектора, информацию о дивидендах. Схема разработанной базы данных представлена на рисунке 2.12.

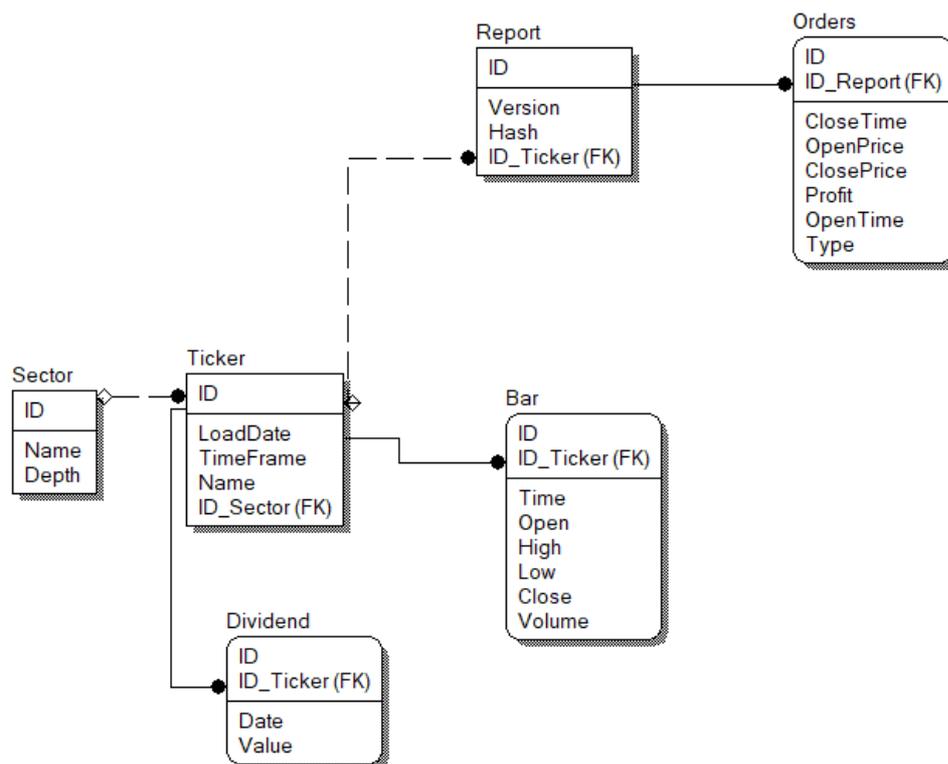


Рисунок 2.12 – База данных для хранения данных

Целесообразно провести индексирование, для ускорения работы базы данных, так как в базе данных может храниться одновременно несколько сотен финансовых инструментов со значительным количеством свечей. Так как база данных служит для ускорения использования финансовых инструментов, то индексирование необходимо выполнять для ключа ID в таблице Bar.

Выводы по второму разделу: были разработаны блок-схемы алгоритмов для описания оптимизации параметров торговой стратегии средствами генетических алгоритмов, а также модели потоков данных DFD информационной системы. Кроме того, была разработана модель базы данных для хранения необходимой информации о стратегиях и результатах тестирования, а также диаграмма классов UML системы оптимизации торговой стратегии.

2 Описание разработанной автоматизированной системы поиска оптимальной торговой стратегии для рынка ценных бумаг

3.1 Описание разработанного пользовательского интерфейса

Для ввода данных и отображений результатов было создано графическое приложение с помощью интерфейса программирования приложений Windows Form. Главное окно изображено на рисунке 3.1.

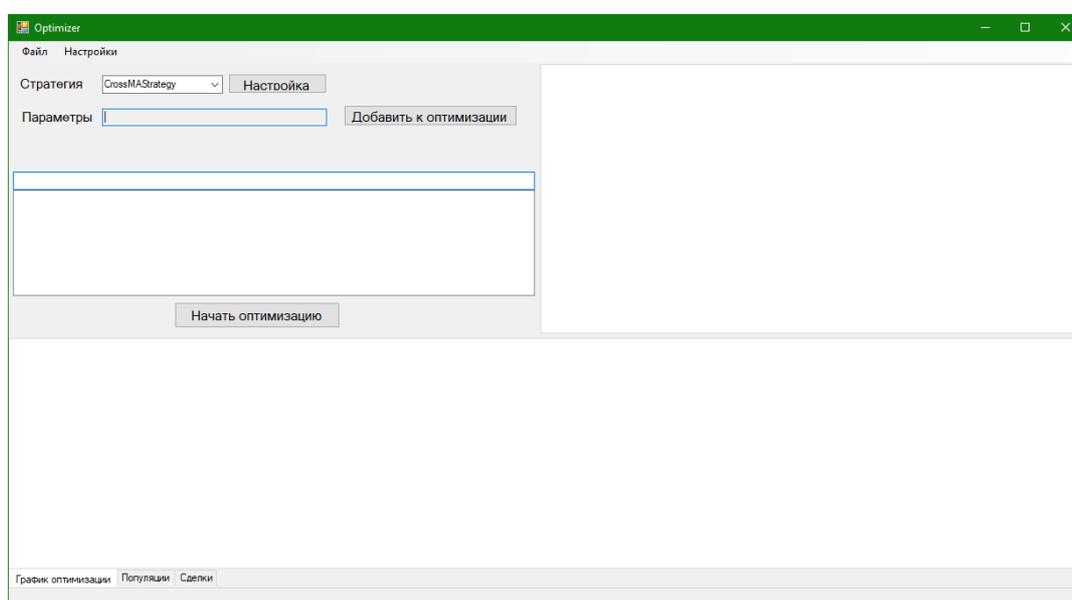


Рисунок 3.1 – Главное окно программы

Основное окно состоит из следующих частей: полосы меню, панель с настройками интерфейса, панель с графиком, на котором изображены сделки выбранной стратегии, панель с результатами оптимизации. Для удобства пользователя программой элементы находятся в контейнере SplitContainer, что позволяет изменять размер графиков и таблиц.

После начала работы с программой необходимо подключиться к базе данных, для этого в меню «Настройки» необходимо выбрать «Подключение к БД» в котором можно будет настроить параметры подключения к базе данных, которые необходимы для работы оптимизатора стратегий.

Окно параметров подключения к базе данных представлено на рисунке 3.2.

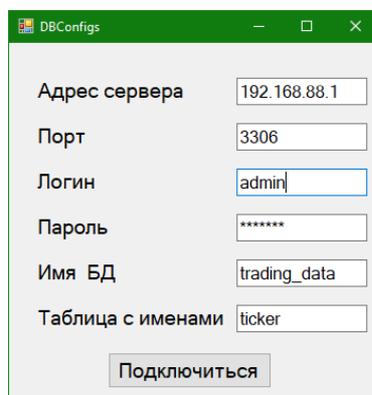
The image shows a window titled "DBConfigs" with a green header bar. It contains several input fields for database connection parameters: "Адрес сервера" (192.168.88.1), "Порт" (3306), "Логин" (admin), "Пароль" (masked with asterisks), "Имя БД" (trading_data), and "Таблица с именами" (ticker). A "Подключиться" button is located at the bottom.

Рисунок 3.2 – Форма с параметрами подключения таблиц

Чтобы добавить стратегию для оптимизации пользователю необходимо выбрать ее название из списка доступных стратегий.

Раскрытое меню со списком доступных стратегий представлено на рисунке 3.3.

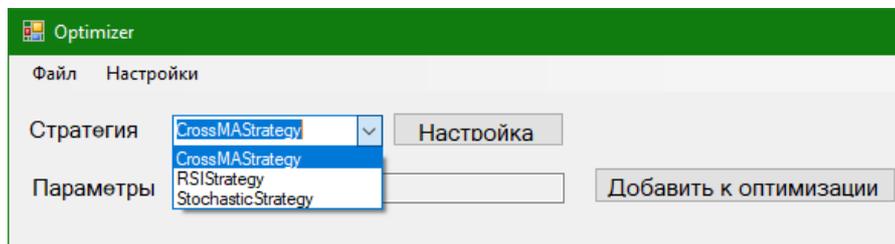
The image shows a window titled "Optimizer" with a green header bar. It has a menu bar with "Файл" and "Настройки". Below the menu bar, there is a "Стратегия" dropdown menu with a list of strategies: "CrossMAStrategy", "CrossMAStrategy", "RSIStrategy", and "StochasticStrategy". To the right of the dropdown is a "Настройка" button. Below the dropdown is a "Параметры" label and an empty input field. To the right of the input field is a "Добавить к оптимизации" button.

Рисунок 3.3 – Список доступных стратегий на главной форме

Чтобы после написания собственной стратегии ее название и отобразилось списке стратегий на форме, тип стратегии необходимо добавить в статический конструктор класса StrategiesList.

Для добавления торговых инструментов к выбранной стратегии, а также для настройки ее параметров необходимо нажать кнопку «Настроить», после чего появится модальное окно настройки стратегии, в заголовке которой отобразится название редактируемой стратегии.

На рисунке 3.4 представлена форма для настройки параметров торговой стратегии.

The screenshot shows a software window titled "StratForm Strategies.CrossMAStrategy". It contains several input fields and buttons for configuring a trading strategy. At the top, there are two date pickers: "Дата начала" (Start Date) set to "1 января 2010 г." and "Дата окончания" (End Date) set to "10 марта 2018 г.". Below these are two dropdown menus for "Добавить инструмент" (Add Instrument) and "Выбранные инструменты" (Selected Instruments), each with a corresponding "Добавить" (Add) and "Удалить" (Remove) button. A table titled "Параметры" (Parameters) is located in the center, with columns for "Название" (Name), "Значение" (Value), "Начало" (Start), "Шаг" (Step), and "Стоп" (Stop). The table contains two rows: "periodFastMA" with a value of 8, and "periodSlowMA" with a value of 14. A "Применить" (Apply) button is at the bottom.

	Название	Значение	Начало	Шаг	Стоп
▶	periodFastMA	8	6	1	25
	periodSlowMA	14	6	1	25

Рисунок 3.4 – Форма настроек параметров торговой стратегии

С помощью данной формы необходимо указать даты начала и окончания тестирования, добавить инструменты в стратегию. При необходимости возможно изменить параметры, редактировать список инструментов в стратегии. Для изменения значений параметра необходимо поменять данные в ячейке таблицы.

Форма с измененными параметрами и раскрытым списком «Добавить инструмент» изображена на рисунке 3.5.

Название	Значение
periodFastMA	4
periodSlowMA	7

Стоп
10
10

Рисунок 3.5 – Форма с открытым списком инструментов и измененными параметрами

Если все параметры корректны, и торговые инструменты были добавлены, то при нажатии кнопки «Применить» параметры стратегии будут применены и окно закроется, иначе будет вызвано диалоговое окно с пояснением причины невозможности добавления указанных данных в параметр.

После нажатия кнопки «Применить», в случае успешного изменений значений параметра, форма закрывается и данные о стратегии помещаются в не редактируемое текстовое поле. После нажатия кнопки «Добавить к оптимизации» данные из поля помещаются в список для оптимизации.

Результат добавления данных о стратегии к оптимизации представлен на рисунке 3.6.

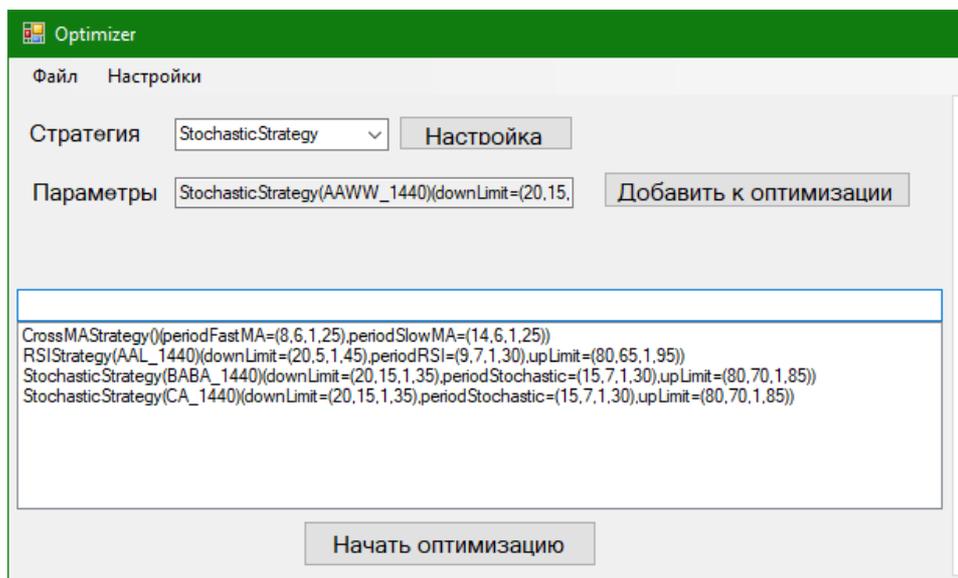


Рисунок 3.6 – Список стратегий с параметрами для оптимизации

Для настроек параметров оптимизации необходимо выбрать меню «Настройки» и «Настройки оптимизации». После чего появится возможность выбора количества особей в популяции, вероятность применения оператора селекции, вероятность применения оператора мутации и вероятность применения оператора инверсии.

На рисунке 3.7 представлено окно с п меню «Настройки» и «Настройки оптимизации».

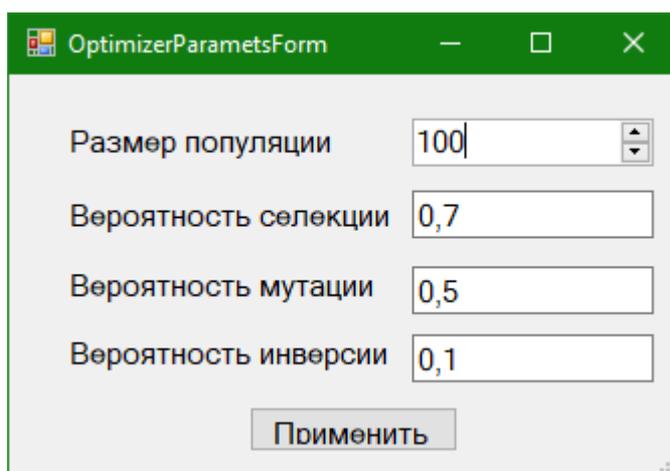


Рисунок 3.7 – Окно настроек параметров

Для начала оптимизации необходимо нажать кнопку «Начать оптимизацию». После выполнения оптимизации в нижней части параметра отобразятся сведения о результатах тестирования.

На вкладке график оптимизации появится график, по оси абсцисс обозначен номер эпохи популяции, по оси ординат – значение фитнес функции.

Пример графика оптимизации из семи популяций представлен на рисунке 3.8.

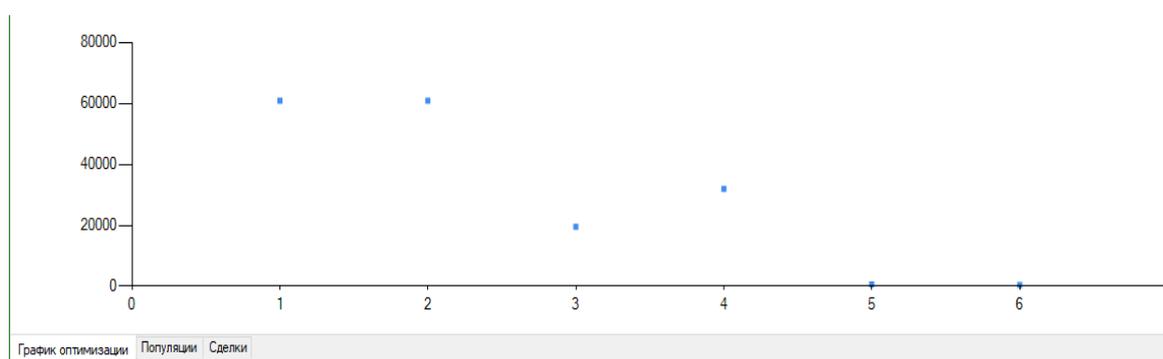


Рисунок 3.8 – График с результатами оптимизации

На вкладке «Популяции», в таблице представлены результаты тестирования стратегий из указанной популяции. В столбце «Стратегия» указывается название стратегии, в столбце результат – мера приспособленности индивида и в столбце «Параметры» – полный список инструментов и параметров, которые используются в стратегии.

При нажатии кнопки «Открыть график» в верхнюю область добавляется график, визуализирующий ордера на исторических данных используемого торгового инструмента.

На рисунке 3.9 изображена информация о торговых стратегиях из первой популяции и представлен график сделок одной из стратегий.

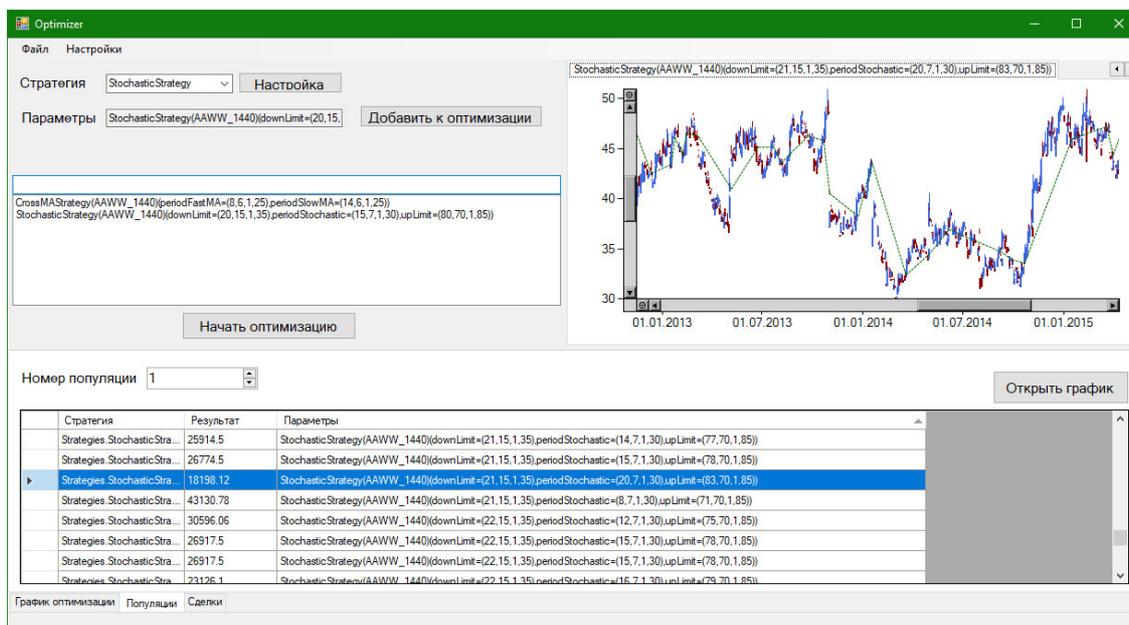


Рисунок 3.9 – Информация о стратегиях в первой популяции

В третьей вкладке под названием «Сделки» отображены все сделки после тестирования выбранной стратегии на вкладке «Популяции». В таблице отображены вся информация о каждой сделке, такие как: тикет (номер сделки), символ (название инструмента), тип (тип сделки, один из Buy, Sell, BuyLimit, SellLimit, BuyStop, SellStop), время открытия и закрытия, цена открытия и закрытия, уровень stoploss и TakeProfit, а также комментарий при открытии сделки и причина закрытия.

На рисунке 3.10 представлено общее окно программного кода, со сделками выбранной стратегии и их изображение на ценовом графике.

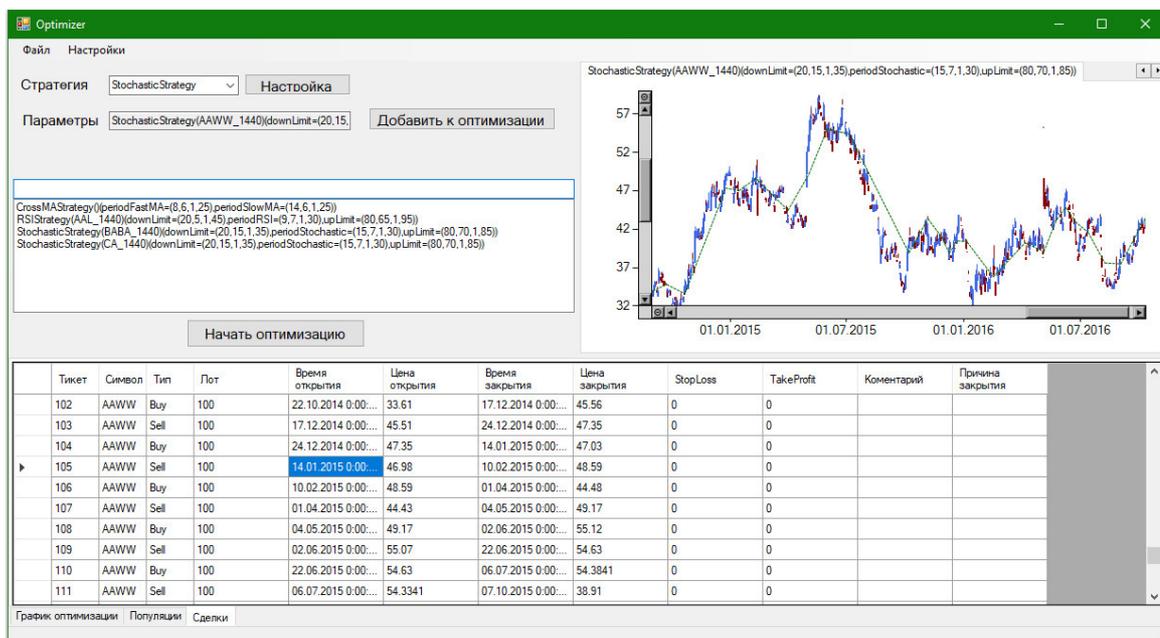


Рисунок 3.10 – Сделки стратегии и соответствующий график

Для отображения биржевых данных используется график с японскими свечами, «бычьи» свечи (цены закрытия в которых выше цен открытия) обозначены синим цветом, а «медвежьи» (цены открытия которых выше цен закрытия) обозначены красным цветом. Для графика цен имеется возможность изменять масштаб, изменять в окне отображаемый временной интервал.

Чтобы изменить масштаб необходимо выделить интересующую область графика, как представлено на рисунке 3.11.

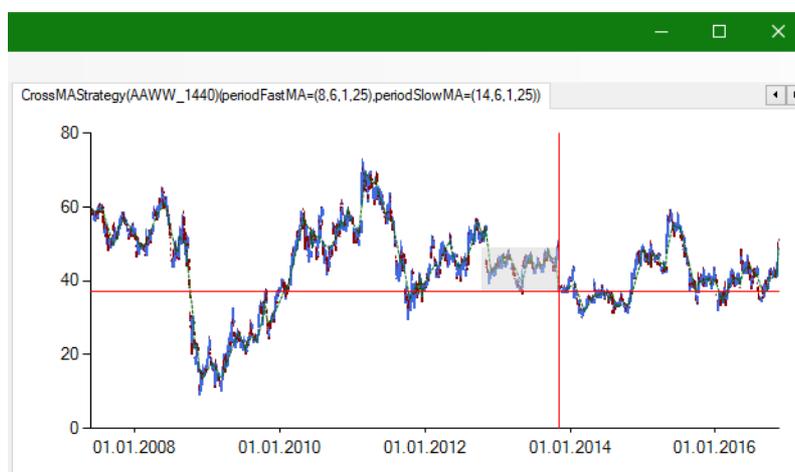


Рисунок 3.11 – Изменение масштаба

После этого масштаб будет изменен и появятся ленты прокручивания, для изменения отображаемого временного интервала.

На рисунке 3.12 изображен график с измененным масштабом.

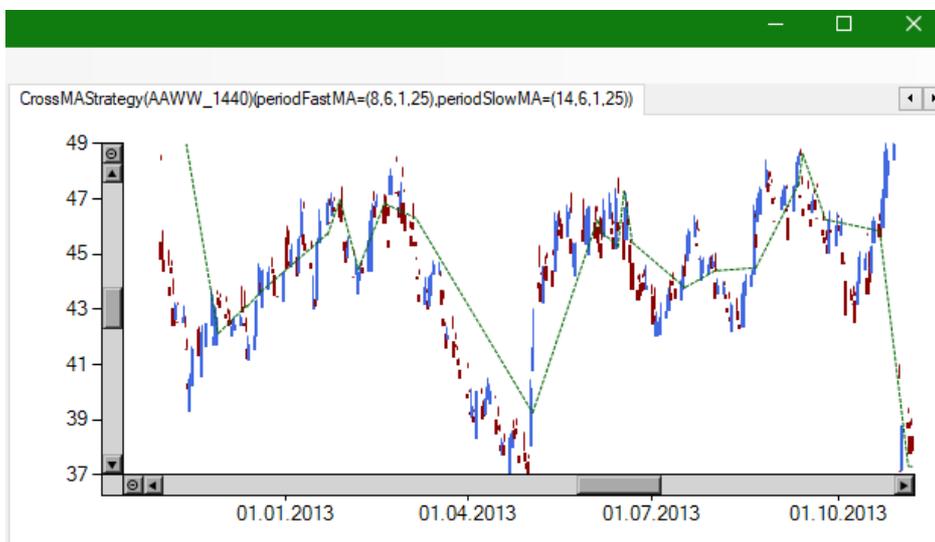


Рисунок 3.12 – График с измененным масштабом

Для возвращения графика в исходное состояние, необходимо нажать на кнопки рядом с полосами прокрутки, что приведет к изменению масштаба на исходное по соответствующим осям графика. На рисунке 3.13 стрелками показано расположение кнопок изменения масштаба.



Рисунок 3.13 – Расположение кнопок изменения масштаба на графике

Таким образом был реализован интерфейс программы, для ввода данных, изменения параметров стратегий и параметров оптимизации, код программы работы основных классов представлен в приложении.

3.2 Поиск оптимальных параметров торговой стратегии с помощью разработанной информационной системы

Для выполнение тестирования контрольного примера использовалось стратегии CrossMAStrategy и StochasticStrategy, в которых использовался инструмент AAWW (котировки акций компании «Atlas Air Worldwide Holdings») и CA (котировки акций компании «CA Technologies») сделки по которым выполняются в NASDAQ — американской бирже, которая специализируется на акциях высокотехнологичных компаний.

На рисунке 3.14 представлены параметры стратегий для оптимизации, а также настройки параметров генетического алгоритма.

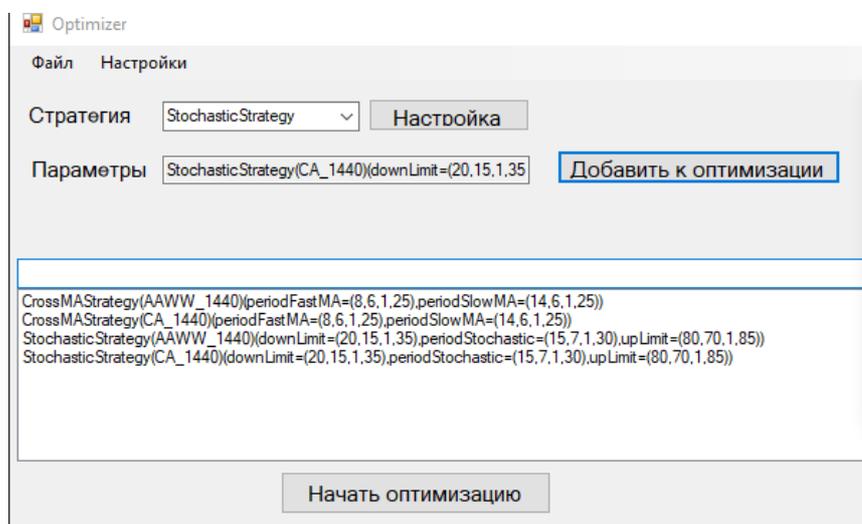


Рисунок 3.14 – Исходные параметры для оптимизации стратегии

После добавления параметров были изменены стандартные значения параметров оптимизации, на значения, представленные на рисунке 3.15.

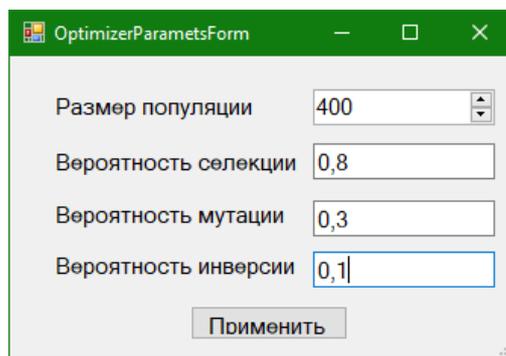


Рисунок 3.15 – Параметры оптимизации

Размер популяции составляет 400 индивидов, это означает, что при генерации первой популяции создастся 100 индивидов каждого типа стратегий.

График оптимизации из 12 значений фитнес функций популяций представлен на рисунке 3.16.

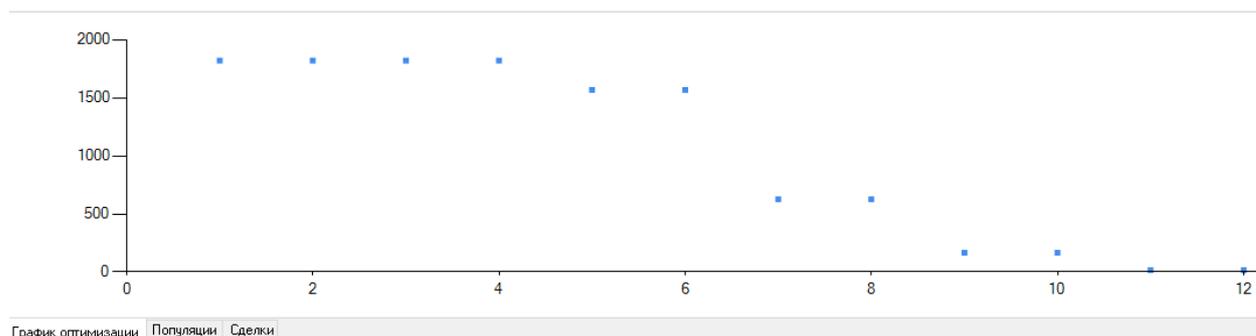


Рисунок 3.16 – График оптимизации

Так как выполнилось условие завершения, это означает, что с ростом числа номера популяции разница между лучшими стратегиями стала меньше заданной точности.

Таким образом с помощью генетического алгоритма было исследовано $400 \cdot 12 = 4800$ наборов параметров стратегий.

Из 400 стратегий в последней популяции лучшими характеристиками обладают следующие стратегии, параметры которых представлены на рисунке 3.17.

Стратегия	Результат	Параметры
Strategies.Stoch...	47029.00	StochasticStrategy(AAWW_1440)(downLimit=(32,15,1,35),periodStochastic=(11,7,1,30),upLimit=(74,70,1,85))
Strategies.Stoch...	43904.00	StochasticStrategy(AAWW_1440)(downLimit=(15,15,1,35),periodStochastic=(21,7,1,30),upLimit=(70,70,1,85))
Strategies.Stoch...	24338.00	StochasticStrategy(CA_1440)(downLimit=(15,15,1,35),periodStochastic=(7,7,1,30),upLimit=(70,70,1,85))
Strategies.Stoch...	23934.00	StochasticStrategy(CA_1440)(downLimit=(20,15,1,35),periodStochastic=(6,7,1,30),upLimit=(80,70,1,85))
Strategies.Stoch...	23908.00	StochasticStrategy(CA_1440)(downLimit=(16,15,1,35),periodStochastic=(7,7,1,30),upLimit=(71,70,1,85))
Strategies.Stoch...	23901.00	StochasticStrategy(CA_1440)(downLimit=(20,15,1,35),periodStochastic=(7,7,1,30),upLimit=(72,70,1,85))
Strategies.Stoch...	23858.00	StochasticStrategy(CA_1440)(downLimit=(20,15,1,35),periodStochastic=(8,7,1,30),upLimit=(74,70,1,85))
Strategies.Stoch...	23338.00	StochasticStrategy(CA_1440)(downLimit=(20,15,1,35),periodStochastic=(8,7,1,30),upLimit=(75,70,1,85))
Strategies.Stoch...	23338.00	StochasticStrategy(CA_1440)(downLimit=(20,15,1,35),periodStochastic=(8,7,1,30),upLimit=(80,70,1,85))
Strategies.Stoch...	23338.00	StochasticStrategy(CA_1440)(downLimit=(20,15,1,35),periodStochastic=(7,7,1,30),upLimit=(80,70,1,85))

Рисунок 3.17 – Лучшие стратегии последней популяции

Наибольшей эффективности по выбранному критерию получила стратегия StochasticStrategy с использованием символа AAWW и следующими значениями параметров downLimit=32, periodStochastic=11, upLimit=74.

Исходя из полученных на рисунке 13 результатов можно получить информацию о сделках лучшей стратегии. Для исследуемого варианта было получены данные о 188 сделках, часть из которых представлены ниже. На рисунке 3.18 представлена информация о первых десяти сделках лучшей стратегии.

Тикет	Символ	Тип	Лот	Время открытия	Цена открытия	Время закрытия	Цена закрытия
1	AAWW	Sell	100	16.05.2007 0:00:...	58.86	22.05.2007 0:00:...	58.76
2	AAWW	Buy	100	22.05.2007 0:00:...	58.76	25.05.2007 0:00:...	58.41
3	AAWW	Sell	100	25.05.2007 0:00:...	58.36	18.06.2007 0:00:...	57.92
4	AAWW	Buy	100	18.06.2007 0:00:...	57.92	24.07.2007 0:00:...	59.62
5	AAWW	Sell	100	24.07.2007 0:00:...	59.57	06.09.2007 0:00:...	50.63
6	AAWW	Buy	100	06.09.2007 0:00:...	50.63	01.10.2007 0:00:...	52.7976
7	AAWW	Sell	100	01.10.2007 0:00:...	52.7476	09.10.2007 0:00:...	53.82
8	AAWW	Buy	100	09.10.2007 0:00:...	53.82	07.11.2007 0:00:...	57.64
9	AAWW	Sell	100	07.11.2007 0:00:...	57.59	07.12.2007 0:00:...	53.1
10	AAWW	Buy	100	07.12.2007 0:00:...	53.1	17.12.2007 0:00:...	53.41
11	AAWW	Sell	100	17.12.2007 0:00:...	53.36	21.12.2007 0:00:...	53.3
12	AAWW	Buy	100	21.12.2007 0:00:...	53.3	03.01.2008 0:00:...	53.89

Рисунок 3.18 – Информация о первых десяти сделках

На рисунке 3.19 представлена информация о последних сделках лучшей стратегии.

Тикет	Символ	Тип	Лот	Время открытия	Цена открытия	Время закрытия	Цена закрытия
177	AAWW	Sell	100	16.06.2016 0:00:...	44.48	11.07.2016 0:00:...	40.37
178	AAWW	Buy	100	11.07.2016 0:00:...	40.37	26.07.2016 0:00:...	41.92
179	AAWW	Sell	100	26.07.2016 0:00:...	41.87	01.08.2016 0:00:...	42.33
180	AAWW	Buy	100	01.08.2016 0:00:...	42.33	03.08.2016 0:00:...	43.77
181	AAWW	Sell	100	03.08.2016 0:00:...	43.72	24.08.2016 0:00:...	35.02
182	AAWW	Buy	100	24.08.2016 0:00:...	35.02	15.09.2016 0:00:...	38.01
183	AAWW	Sell	100	15.09.2016 0:00:...	37.98	23.09.2016 0:00:...	39.86
184	AAWW	Buy	100	23.09.2016 0:00:...	39.86	13.10.2016 0:00:...	42.84
185	AAWW	Sell	100	13.10.2016 0:00:...	42.79	25.10.2016 0:00:...	42.6
186	AAWW	Buy	100	25.10.2016 0:00:...	42.6	03.11.2016 0:00:...	41.85
187	AAWW	Sell	100	03.11.2016 0:00:...	41.8	07.11.2016 0:00:...	42.65
188	AAWW	Buy	100	07.11.2016 0:00:...	42.65	15.11.2016 0:00:...	48.95

Рисунок 3.19 – Последняя десятка стратегии

На рисунке 3.20 изображен полученный график с последними сделками наиболее эффективной стратегии.



Рисунок 3.20 – Последние сделки стратегии на графике инструмента

Чтобы посчитать число возможных вариантов наборов значений параметров при полном переборе, необходимо вычесть из максимального значение параметра минимальное и разделить на величину шага, после этого необходимо найти произведение полученных величин.

Таблица 3.1 – Параметры стратегий с количеством вариантов возможных наборов

Параметры стратегии	Количество вариантов
CrossMAStrategy(AAWW_1440) (periodFastMA=(8,6,1,25), periodSlowMA=(14,6,1,25))	361
CrossMAStrategy(CA_1440) (periodFastMA=(8,6,1,25), periodSlowMA=(14,6,1,25))	361
StochasticStrategy(AAWW_1440) (downLimit=(20,15,1,35), periodStochastic=(15,7,1,30), upLimit=(80,70,1,85))	8100
StochasticStrategy(CA_1440) (downLimit=(20,15,1,35), periodStochastic=(15,7,1,30), upLimit=(80,70,1,85))	8100

Таким образом всего существует 16922 возможных вариантов значений параметров, это означает что количество исследуемых вариантов при использовании разработанной модели сократилось более чем в 3.5 раза (до 4800 наборов). На рисунке 3.21 изображен график со значениями выбранного критерия.

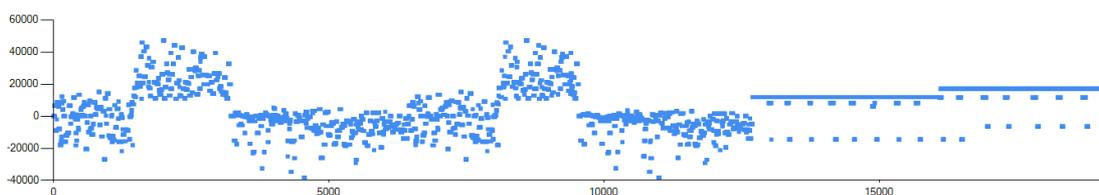


Рисунок 3.21– График результатов при полном переборе параметров

Выводы по третьему разделу: в данном разделе описан разработанный пользовательский интерфейс автоматизированной системы поиска оптимальной торговой стратегии для рынка ценных бумаг, а также описан контрольный пример реализации проекта.

ЗАКЛЮЧЕНИЕ

В большинстве случаев трейдеры скрывают свои разработки и средства моделирования и оптимизация поведения торговой стратегии для рынка ценных бумаг. Разработанная информационная система и оригинальный метод оптимизации с использованием генетического алгоритма реализует большую часть функционала, необходимого для автоматизации деятельности трейдера при оптимизации параметров торговой стратегии для рынка ценных бумаг.

В процессе выполнения магистерской диссертации в первом разделе было проведено общее исследование процесса разработки торговой стратегии и описаны принципы оценки их эффективности. Проведен анализ существующих аналогичных информационных систем и выявлены их недостатки. Обоснована необходимость в разработке новой модели поиска оптимальной торговой стратегии. Проведена общая характеристика организации решения задачи.

Во втором разделе описаны виды исследуемых торговых стратегий. Были рассмотрены основные этапы и понятия классического генетического алгоритма. На основании проведенного анализа была разработана модель поиска оптимальной торговой стратегии средствами генетического алгоритма для сокращения количества исследуемых вариантов. Для реализации разработанной модели было принято решение о необходимости разработки новой информационной системы с обоснованием требований, выбором архитектуры приложения и средства реализации. Выполнено проектирование автоматизированной системы с применением методологии DFD, с разработкой контекстной диаграммы и выполнением декомпозиции. Создана диаграмма классов в соответствии с нотацией UML и разработана схема базы данных для хранения сведений о инструментах, отчетах, котировках, секторах.

В третьем разделе представлено описание графического интерфейса пользователя разработанной информационной системы с подробным

описание функционала используемых элементов. Программное приложение позволяет строить графики оптимизации, а также выполнять поиск наилучших параметров стратегий с предоставлением подробной информации о исследуемых торговых стратегиях и смоделированных сделках. Проведено сравнение количества исследуемых наборов параметров стратегий с использованием разработанной модели, основанной на генетическом алгоритме, и методом полного перебора.

Таким образом, все поставленные задачи выполнены. По результатам исследования и проектирования была реализована автоматизированная система поиска оптимальной торговой стратегии для рынка ценных бумаг. Для разработки системы использовалась современная среда разработки Microsoft Visual Studio 2017, что позволило быстро создать пользовательский интерфейс, вести удобную и качественную разработку, без применения дополнительных решений.

Использование разработанной системы позволит снизить временные затраты на поиски оптимальной торговой стратегии, автоматизировать рутинные действия пользователя.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Швагер, Джек Д.. Технический анализ. Полный курс / Джек Д. Швагер – М.: Альпина Паблишер, 2017. – 880 с.
2. Мэрфи, Джон Дж.. Технический анализ фьючерсных рынков. Теория и практика / Джон Дж. Мэрфи – М.: Альпина Паблишер, 2011. – 616 с.
3. Черноморец, А.А. Информационные системы валютного рынка / А.А. Черноморец: Белгородский институт государственного и муниципального управления ОРАГС, 2006 – 173 с.
4. Киселевич, Ю.В. Применение моделей оценки конкурентоспособности при формировании оптимального портфеля ценных бумаг //Материалы международной научно-практической конференции «Конкурентоспособность бизнеса и технологий» / Ю.В. Киселевич - Брянск, 2005
5. Вдовин, В.М. Информационные технологии в финансово-банковской сфере: Практикум / В.М. Вдовин, Л.Е Суркова - Москва: Дашков и К°, 2010. – 248 с.
6. Гахов, Р.П. Компьютерное моделирование экономических процессов: учебное пособие для студентов вузов по специальности 230400.62 "Информационные системы и технологии" [Текст] / Р.П.Гахов, Н.В. Щербинина и др. - Белгород: ИД Белгород, 2014. – 88 с.
7. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский – М.: Горячая линия – Телеком, 2006. – 452 с.
8. Воеводин В.В.: Вычислительная математика и структура алгоритмов. - М.: Московский университет, 2010. – 157 с.
9. Мезенцев, К.Н. Автоматизированные информационные системы / К.Н. Мезенцев. – М.: Академия, 2012. – 174 с.

10. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер, Д. Райс, М. Фоммел, Э. Хайет, Р. Ми, Р. Стаффорд – М.: Вильямс, 2007. – 544 с.
11. Коберн А. Быстрая разработка программного обеспечения / А. Коберн. – М.: Лори, 2013 – 336 с.
12. Хорев, П. Б. Объектно-ориентированное программирование: [учебное пособие по направлению "Информатика и вычислительная техника"] / П. Б. Хорев. - М., 2011. – 446, [1] с. : ил. - Рекомендовано УМО.
13. Троелсен, Э. Язык программирования C# 6.0 и платформа .NET 4.6 / Э. Троелсен. – Вильямс, 2016. – 1440 с.
14. Рихтер, Дж. C# Программирование на платформа NET / Дж. Рихтер. - СПб.: Питер, 2012
15. Нейгел, К. C# 2008 и платформа NET для профессионалов/ К. Нейгел, Б. Ивѐн Б. и др. - М.: Диалектика, 2008
16. Линев А.В.: Технологии параллельного программирования для процессоров новых архитектур. - М.: Московский университет, 2010
17. Гахов, Р.П. Компьютерное моделирование экономических процессов [Текст]: учебное пособие для студентов вузов / Р.П. Гахов, Н.В. Щербинина - Белгород: ИД Белгород, 2014. – 88 с.
18. Гусятников, В.Н. Стандартизация и разработка программных систем: Учебное пособие: [Электронный ресурс]/ В.Н. Гусятников, А.И. Безруков; В.Н. Гусятников, А.И. Безруков. - Москва : Финансы и статистика : Инфра-М, 2014. – 288 с.
19. Талеб, Н.Н. Одураченные случайностью. О скрытой роли шанса в бизнесе и в жизни / Н.Н. Талеб – М.: Манн, Иванов и Фебер, 2016. – 320 с.
20. Румбешт, В.В. Программная инженерия [Электронный ресурс]: учебно-методический комплекс / В. В. Румбешт - Белгород, 2013.
21. Ломакин, В.В. Программирование и программное обеспечение информационных технологий [Электронный ресурс] / В.В. Ломакин – НИУ БелГУ. - Белгород, 2014.

22. Петина, М.А. Инструментальные средства поддержки жизненного цикла программно-информационных средств [Электронный ресурс]: Учебно-методический комплекс / М.А. Петина, А.Н. Коваленко, Т.Г. Кузьмичева. - Белгород, 2017. – 165 с.

23. Маторин, С.И. Теория систем и системный анализ [Электронный ресурс] / С.И. Маторин, О.А. Зимовец - Белгород: ИД Белгород, 2015.

24. Гахов, Р.П. Методы и средства проектирования информационных систем и технологий: Учебно-методический комплекс [Электронный ресурс] / Р.П. Гахов – НИУ БелГУ. - Белгород, 2013.

25. Душин, В.К. Теоритические основы информационных процессов и систем / В.К Душин - М.: Дашков и К, 2010

26. Иорш, В.И. Управление основными фондами на основе ключевых показателей эффективности [Текст] / В. И. Иорш, В. Д. Стружинский // Горный журнал. – 2010. - №3. – С. 25 – 28.

27. Уоррен, Г. С. Алгоритмические трюки для программистов / Г. С. Уоррен. – Вильямс, 2014. – 512 с.

28. Власов Д.В., Божко В.П. - Информационные технологии в экономике и управлении: Учебно-методический комплекс - Москва: Евразийский открытый институт, 2010. - 167 с.

29. ГОСТ 7.32-2001 Отчет о научно-исследовательской работе. Структура и правила оформления [Текст]. - М.: Изд-во стандартов, сор. 2001. – 26 с. - (Система стандартов по информации, библиотечному и издательскому делу).

30. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание [Текст]. - М.: Изд-во стандартов, сор. 2004. – 170 с. - (Система стандартов по информации, библиотечному и издательскому делу).

ПРИЛОЖЕНИЕ

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Terminal;

namespace Optimizer
{
    public class Optimizer
    {
        #region params

        private int randomIniter;
        static public int populationSize = 10;

        #endregion

        private List<Terminal.Terminal> strats;

        public List<Population> PopulationList { get; private set; }

        public Optimizer(IEnumerable<Terminal.Terminal> strats)
        {
            this.strats = strats.ToList();
            PopulationList = new List<Population>(1000);
            var startPopulation = new Population(strats);
            PopulationList.Add(startPopulation);
        }

        protected bool StopCondition()
        {
            bool res = true;
            int lastIndex = PopulationList.Count - 1;

            if (lastIndex < 5) {
                return false;
            }

            if (lastIndex > 4) {
                return true;
            }

            var eps = 100.0;
            var diff = PopulationList[lastIndex].Fitnes - PopulationList[lastIndex -1].Fitnes;
            diff = Math.Abs(diff);
            if (diff < eps) {
                return true;
            }

            return false;
        }

        public void Run()
        {
            for (Population current = PopulationList.First();
                !StopCondition();
```

```

        current = new Population(current)) {
            current.Run();
            PopulationList.Add(current);
        }
    }
}
}
}

```

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;
using Terminal;

```

```

namespace Optimizer
{
    public class Population : IEnumerable<Individ>
    {
        int epoch = 0;
        public int countOfStratsInGroup = 100;
        public int randomIniter = 0;

        protected List<Individ> individs;
        private Dictionary<Individ, double> profits;
        private Population oldPopulation;
        private List<Group> groupList;

        public double Fitness { get; private set; }

        public Population(IEnumerable<Terminal.Terminal> strats)
        {
            int count = strats.Count();
            groupList = new List<Group>(count);
            individs = new List<Individ>(count * countOfStratsInGroup);
            this.profits = new Dictionary<Individ, double>(count);

            foreach (var strat in strats)
            {
                var temp = new Group(strat, countOfStratsInGroup);
                groupList.Add(temp);
                individs.AddRange(temp.individs);
            }
        }

        public Population(Population oldPopulation)
        {
            this.oldPopulation = oldPopulation;

            epoch = oldPopulation.epoch + 1;
            int count = oldPopulation.groupList.Count();

```

```

    groupList = new List<Group>(count);
    individs = new List<Individ>(count * countOfStratsInGroup);
    this.profits = new Dictionary<Individ, double>(count);

    MethodSelector(oldPopulation);
}

protected void MethodSelector(Population oldPopulation)
{
    MethodSelector(oldPopulation.individs, oldPopulation.Fitnes);
}

void AddIndivid(Individ target)
{
    individs.Add(target);

    //adding to grouplist
    bool isNewGroupneed = true;
    foreach (var targetGroup in groupList) {
        if (targetGroup.IsParentGroup(target.OwnerGroup)) {
            isNewGroupneed = false;
            target.OwnerGroup = targetGroup;
            targetGroup.AddIndivid(target);
        }
    }
    //create new group and adding to grouplist
    if (isNewGroupneed) {
        var temp = new Group(target.OwnerGroup);
        temp.AddIndivid(target);
        target.OwnerGroup = temp;
        groupList.Add(temp);
    }
}

protected void MethodSelector(List<Individ> targetIndivids, double sum)
{
    int count = targetIndivids.Count;

    double rouletteArrow = -1;

    var mainSelector = new Selector(oldPopulation.individs);
    for (int i = 0; i < count; ++i) {

        Individ parentIndivid = mainSelector.GetNext();
        Individ childIndivid = null;

        if (Individ.ProbabilityCrosinger < MyRandom.NextDouble()) {
            Individ parent2 = parentIndivid.OwnerGroup.GetNext();
            childIndivid = parentIndivid.Crossinger(parent2);
        }
        if (Individ.ProbabilityMutation < MyRandom.NextDouble()) {
            childIndivid = parentIndivid.Mutation();
        }
        if (Individ.ProbabilityInversion < MyRandom.NextDouble()) {
            childIndivid = parentIndivid.Inversion();
        }

        if (childIndivid == null) {
            childIndivid = parentIndivid.MakeCopy();
        }

        childIndivid.OwnerGroup = parentIndivid.OwnerGroup;
        AddIndivid(childIndivid);
    }
}

```

```

    }
}

void Prepare(IEnumerable<Terminal.Terminal> strats)
{
}

public double Run()
{
    foreach (var individ in individs) {
        double temp = individ.Run();
        profits.Add(individ, temp);
    }

    Fitness = GetFitness();

    return Fitness;
}

protected double GetFitness()
{
    return individs.Max(x => x.Result) - individs.Average(x=>x.Result);
    //return individs.Select(x => x.Result).OrderBy(x => -x)
    //    .Take((int)(individs.Count * 0.1)).Average();
}

public IEnumerator<Individ> GetEnumerator()
{
    return ((IEnumerable<Individ>)individs).GetEnumerator();
}

IEnumerator IEnumerable.GetEnumerator()
{
    return ((IEnumerable<Individ>)individs).GetEnumerator();
}
}
}

```

```

using System;
using System.CodeDom.Compiler;
using System.Collections;
using System.Collections.Generic;
using System.Data.OleDb;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using Terminal;

```

```

namespace Optimizer
{
    public class Individ
    {
        public int randomIniter = 0;
        public Terminal.Terminal Strat { get; private set; }
        public BitArray Gen { get; private set; }
        public List<int> Counts { get; private set; }
        public int Size { get; private set; }
        public Group OwnerGroup { get; set; }
    }
}

```

```

public double Result
{
    get { return Strat.Result; }
}

public Dictionary<string, Parametr>.ValueCollection Paramtrs;

public static double ProbabilityCrosinger = 0.9;
public static double ProbabilityMutation = 0.7;
public static double ProbabilityInversion = 0.05;

protected Individ()
{
}

public Individ(Terminal.Terminal strat)
{
    Strat = Activator.CreateInstance(strat.GetType()) as Terminal.Terminal;
    Paramtrs = Strat.GetParamtrs().Values;
    foreach (var stratTicker in strat.tickers.Values) {
        Strat.AddSymbol(stratTicker);
    }
    Counts = GetCounts();
    Size = GetSize();
    Gen = new BitArray(Size);
}

public double Run()
{
    return Strat.Run();
}

#region genetic operators

public Individ Crossinger(Individ parent2)
{
    var parent1 = this;
    var child = parent1.MakeCopy();
    int r = MyRandom.Next(Size - 1);

    for (int i = 0; i < r; ++i) {
        child.Gen[i] = parent1.Gen[i];
    }

    for (int i = r; i < Size; ++i) {
        child.Gen[i] = parent2.Gen[i];
    }

    child.Decrypte();

    //if(child.IsNormal()) ??

    return child;
}

public Individ Mutation()
{
    var child = this.MakeCopy();

    int r = MyRandom.Next(Size - 1);

```

```

        child.Gen[r] = !child.Gen[r];

        return child;
    }

    public Individ Inversion()
    {
        var child = this.MakeCopy();
        int r = MyRandom.Next(Size - 1);
        int i, j;
        for (i = 0; r + i < Size; ++i) {
            child.Gen[i] = Gen[r + i];
        }

        for (j = 0; i < Size; ++i, ++j) {
            child.Gen[i] = Gen[j];
        }

        return child;
    }

#endregion

    public Individ MakeCopy()
    {
        var newIndivid = new Individ();

        var newStratTemp = Activator.CreateInstance(Strat.GetType());
        var newStrat = newStratTemp as Terminal.Terminal;

        foreach (var stratTicker in Strat.tickers.Values) {
            newStrat.AddSymbol(stratTicker);
        }
        newStrat.MainSymbol = Strat.MainSymbol;

        newIndivid.Strat = newStrat;

        newIndivid.Size = Size;
        newIndivid.Counts = Counts;
        newIndivid.Gen = (BitArray)(Gen.Clone());
        newIndivid.Parametrs = newIndivid.Strat.GetParametrs().Values;

        return newIndivid;
    }

    public int MakeRandom()
    {
        int countOfTryes = 0;
        do {
            for (int i = 0; i < Size; ++i) {
                Gen[i] = (MyRandom.Next() % 2) == 0;
            }

            countOfTryes++;
            this.Decrypte();
        } while (!IsNormal());

        return countOfTryes;
    }

    public bool IsNormal()
    {
        foreach (var parametr in Strat.GetParametrs().Values) {
            if (parametr.CurrentValue < parametr.StartValue

```

```

        || parametr.CurrentValue > parametr.EndValue) {
            return false;
        }
    }
    return true;
}

private bool GetBit(ref uint target, int pos)
{
    return (((uint)0x1) << pos) & target > 0;
}

private void SetBit(ref uint target, int pos, bool val)
{
    uint temp = (uint)(val ? 1 : 0) << pos;
    target = target | temp;
}
// have params, need gen
public void Encrypte()
{
    int pos = 0;
    int i = 0;

    foreach (var parametr in Strat.GetParams().Values) {
        uint temp = (uint)((parametr.CurrentValue - parametr.StartValue) /
parametr.Step);

        for (int j = Counts[pos] - 1; j >= 0; --j, ++i) {
            Gen[i] = GetBit(ref temp, j);
        }
        i = Counts[pos];
        pos++;
    }
}
//have gen, need params
public void Decrypte()
{
    int pos = 0;
    int i = 0;

    foreach (var parametr in Strat.GetParams().Values) {
        uint temp = 0;

        for (int j = Counts[pos] - 1; j >= 0; --j, ++i) {
            SetBit(ref temp, j, Gen[i]);
        }
        i = Counts[pos];
        parametr.CurrentValue = parametr.StartValue + temp * parametr.Step; // ??
        pos++;
    }
}

List<int> GetCounts()
{
    var res = new List<int>();

    foreach (var parametr in Strat.GetParams().Values) {
        uint temp = parametr.Size;
        int i = 1, k = 2;
        while (k <= temp) {
            ++i;
            k *= 2;
        }
    }
}

```

```

        res.Add(i);
    }

    return res;
}

int GetSize()
{
    return Counts.Sum();
}
}
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Terminal;

namespace Optimizer
{
    public class Selector
    {
        private List<Individ> targetIndivids;

        private double sum;
        private int pos = 1;
        private double rouletteArrow;

        public Selector(List<Individ> individs)
        {
            this.targetIndivids = individs;
            int count = targetIndivids.Count;
        }

        public Individ GetNext()
        {
            Individ res = targetIndivids[0];
            if (pos == 1) {
                sum = GetSum();
                rouletteArrow = MyRandom.NextDouble() * sum;
            }
            double tempRouletteArrow = (rouletteArrow * pos) % sum;
            pos++;
            foreach (var individ in targetIndivids) {
                if (tempRouletteArrow < individ.Result) {
                    return individ;
                } else {
                    if (individ.Result > 0) {
                        tempRouletteArrow -= individ.Result;
                    }
                }
            }

            return res;
        }

        double GetSum()
        {
            return targetIndivids.Sum(x => x.Result > 0 ? x.Result : 0);
        }
    }
}
}

```


Магистерская диссертация выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« ___ » _____ Г.

(подпись)

(Ф.И.О.)