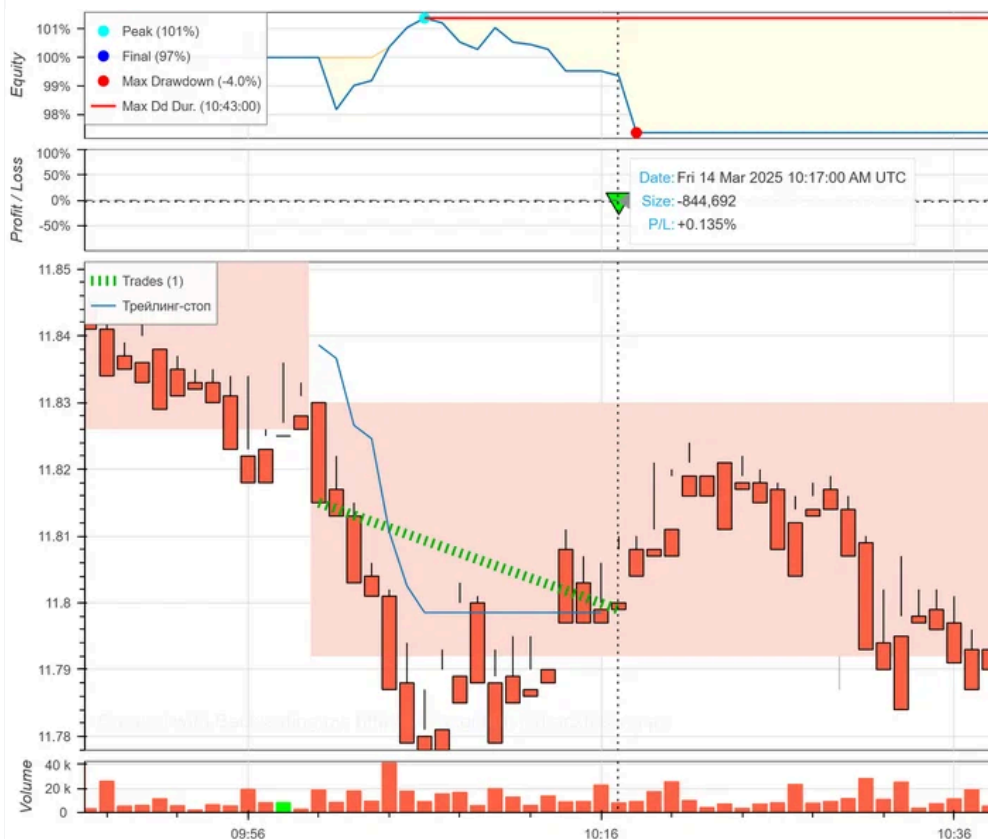


[Горячее](#) [Лучшее](#) [Свежее](#) [Подписки](#) **empenoso** 2 дня назад

Алготрейдинг для начинающих: проверяю на Python работает ли система случайных входов для фьючерсов

💡 Для ЛЛ:

Я протестировал торговую систему для фьючерсов Московской биржи, основанную на идеях Александра Резвякова. Основная суть: сделки открываются в 10:00, используются дневные бары, выход - по времени (20:40) или по трейлинг-стопу. Я реализовал это в Python с библиотекой [backtesting.py](#), код доступен на GitHub. Теперь подробнее для тех, кто хочет разобраться.



 **Воспроизведение и тестирование стратегии Александра Резвякова на фьючерсах**



Используйте аккаунт Яндекса для входа на сервис

Безопасный вход без дополнительной регистрации на сайте

Войти с Яндекс ID

Логин

Пароль

Войти

Создать аккаунт

[Забыли пароль?](#)

или продолжите с



Войти с Яндекс ID



Войти через VK ID

 [Промокоды](#)

 [Работа](#)

 [Курсы](#)

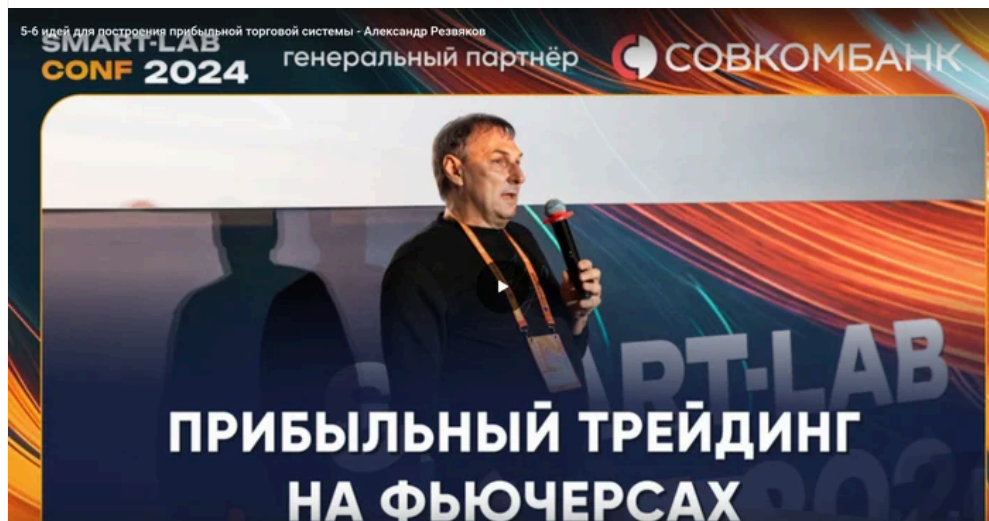
 [Реклама](#)

 [Игры](#)

 [Пополнение Steam](#)



Недавно я наткнулся на выступление Александра Резвякова на конференции 2024 года "5-6 идей для построения прибыльной торговой системы на фьючерсах". Его система меня зацепила своей простотой и логичностью. Я решил воссоздать её, протестировать и проверить, насколько она эффективна.



✓ Основная идея стратегии

1. **Вход** в сделку строго в 10:00.
2. **Лонг**, если за последние два дня минимумы и максимумы росли.
3. **Шорт**, если последние два дня минимумы и максимумы снижались.
4. **Выход** из позиции либо:
 - По времени (20:40);
 - По трейлинг-стопу (движение цены на 2% от максимума для лонга, от минимума для шорта).
5. Дополнительно можно увеличивать объем позиции при успешных сделках.

🌐 Подготовка среды

Для тестирования я использовал **Python** и библиотеку **backtesting.py**. Разбил проект на модули:

- **main.py** – запускает процесс тестирования;
- **resample_data_1min.py** – фильтрует минутные данные из API брокера;
- **data_loader.py** – загружает данные из CSV в нужном формате;
- **backtester.py** – логика бэктеста;
- **strategy_Random_1min.py** – реализация стратегии.

Топ прошлой недели

- Kerbis
2 поста
- MaximVeter
6 постов
- alex.carrier
16 постов

[Посмотреть весь топ](#)

✉ Лучшие посты недели

Рассылка Пикабу
отправляем са
рейтинговые м
дней 🔥



Укажи

Подписаться

Хотя в видео рекомендовали 5-минутные свечи, я взял **1-минутные** – так проще отследить срабатывание трейлинг-стопов.

Реализация стратегии в Python

1. Загрузка данных

Минутные котировки обрабатываются и фильтруются по нужному диапазону.

2. Преобразование данных

CSV-файлы конвертируются в формат, понятный `backtesting.py`.

3. Логика входа/выхода

В 10:00 проверяются два последних дневных бара. Если условия выполняются – вход в лонг/шорт с вероятностью 50%. Выход в 20:40 или по трейлинг-стопу.

➔ Тестирование и результаты

Запуск бэктеста с параметрами:

```
bt = Backtest( df, DynamicStrategyClass, cash=1_000_000, commission=0.002, margin=0.1,
# Маржа 10% trade_on_close=True, hedging=False, )
```

Так как вход случайный (50%), результаты тоже рандомные. Но уже на первых тестах видно, что стратегия даёт интересные закономерности.

Нажимая кнопку «Подписаться на рассылку», я соглашаюсь с [Правилами Пикабу](#) и даю согласие на [обработку персональных данных](#).

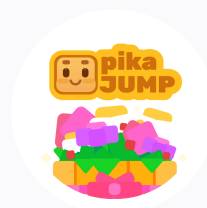
Новости Пикабу
Помощь
Кодекс Пикабу
Реклама
О компании

Команда Пикабу
Награды
Контакты
О проекте
Зал славы

Промокоды
Скидки
Работа
Курсы
Блоги

Купоны Мегамаркет
Купоны Тефаль
Купоны М.Видео
Купоны YandexTravel
Купоны Lamoda

Мобильное приложение



Пример сделки на фьючерсе NGH5:

🚀 Запуск бэктеста для NGH5 ⏳ ПРОДАЖА на 2025-03-12: Цена: 4.33, Стоп: 4.33 ✅

ВЫКУП (трейлинг-стоп) на 2025-03-12: Цена: 4.31

🔧 Код в открытом доступе

Если хотите воспроизвести или улучшить стратегию – код на [GitHub](#). Готов обсудить идеи и улучшения!

Автор: Михаил Шардин

🔗 [Моя онлайн-визитка](#)

📱 [Telegram «Умный Дом Инвестора»](#)

19 марта 2025 г.



💬 6 👁 2.8K 📄 ➦

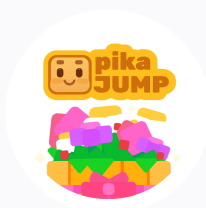
₽ Поддержать

Эмоции

Все комментарии Автора

Раскрыть 6 комментариев

Чтобы оставить комментарий, необходимо [зарегистрироваться](#) или [войти](#)



● ————— ■ —————

—————
—————
—————
—————
—————
—————
—————

● ————— ■ —————

—————
—————
—————
—————
—————
—————
—————

● ————— ■ —————

—————
—————
—————
—————
—————
—————
—————

