

Тестер автоматизированных стратегий FOREX – ТС 1 (Аналог Strategy Tester MT4)

Первая часть

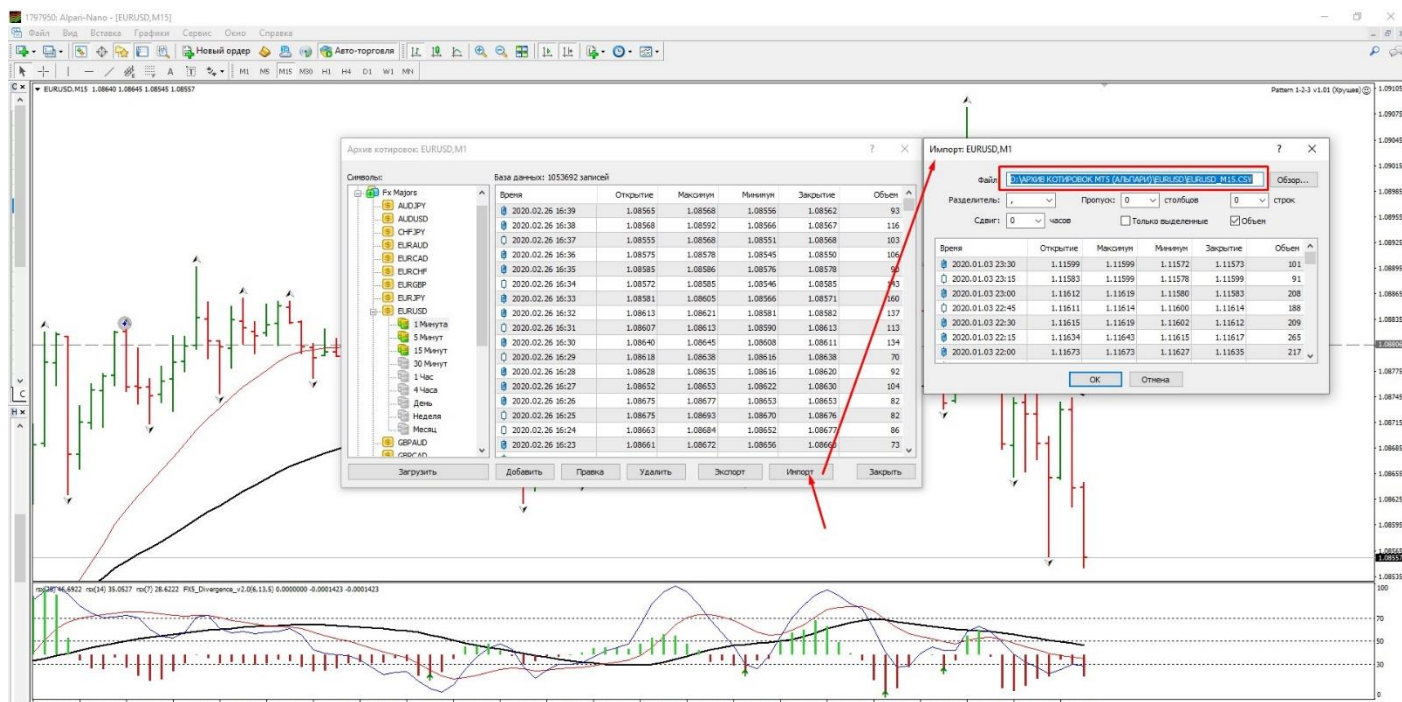
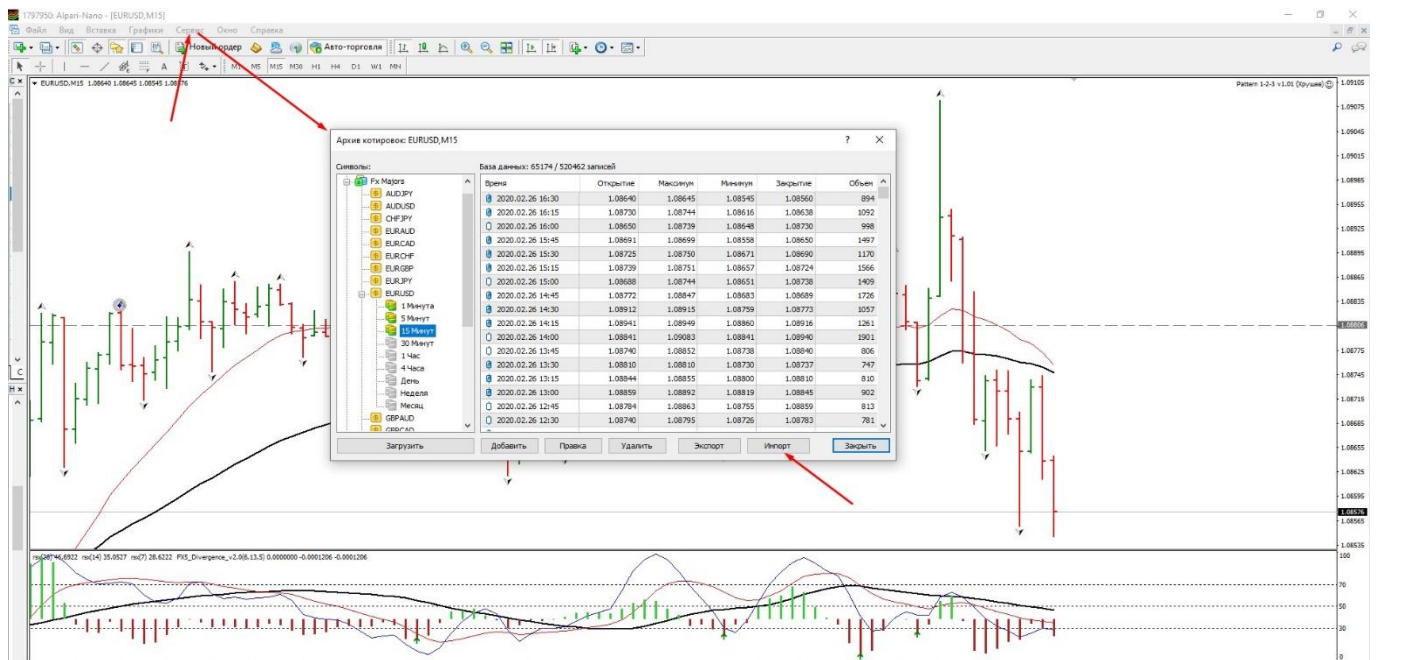
Проблема: за счет того, что стандартная программа для оптимизации (бэк-тест) и тестирования (форвард – тест) торговых роботов – MetaTrader 4 (а большинство торговых роботов написано именно под MT4) в принципе тяжелая и местами просто «деревянная», процесс теста одного алгоритма для одного торгового инструмента (валютной пары, металла и т.д.), а этих инструментов более 50, может занимать 4-6 недель (при минимальных шагах настройки оптимизации) в среднем. Рынок, с течением времени подвержен критическим изменениям, на которые прописанные настройки алгоритма торгового робота просто не смогут отреагировать. Такие изменения происходят примерно раз в 1-1,5 года. Следовательно, получится протестировать один алгоритм для 10-15 инструментов, и его снова нужно переоптимизировать под изменившийся рынок под те же 10-15 инструментов. Да, можно задать больший период оптимизации и получить более универсальные настройки, которые будут работать дольше чем 1 – 1,5 года, но и период такой оптимизации и форвард – тестирования займёт намного больше времени – примерно 6-9 недель. Итого – проблема в скорости самого процесса. Есть вариант тестирования в относительно новом тестере терминала MetaTrader 5. Но если при тестировании даже в одновременно включенных 10 терминалах MT4, 10 разных инструментов – терминал использует лишь одно ядро из имеющихся в железе, то MT5 использует весь потенциал железа даже при тестировании одного торгового инструмента. Да, в MT5 скорость быстрее, но и железо, даже класса upper intermediate «ложится» в самом начале тестирования. Проще говоря – «улучшенный» тестер MT5 не решает временную проблему всего процесса.

Цель: создать отдельную программу, которая будет намного легче (как минимум за счет того, что будет работать отдельно, а не в рамках приложения к тяжёлому торговому терминалу) и как следствие – быстрее, для ускоренной, но в то же время качественной оптимизации (бэк-тест) и тестирования (форвард – тест) торговых роботов для последующей их работы в MetaTrader 4. Настройки, полученные после форвард – теста будут переноситься в MT4 и там проходить проверочные прогоны для большей уверенности в правильности выбора настроек.

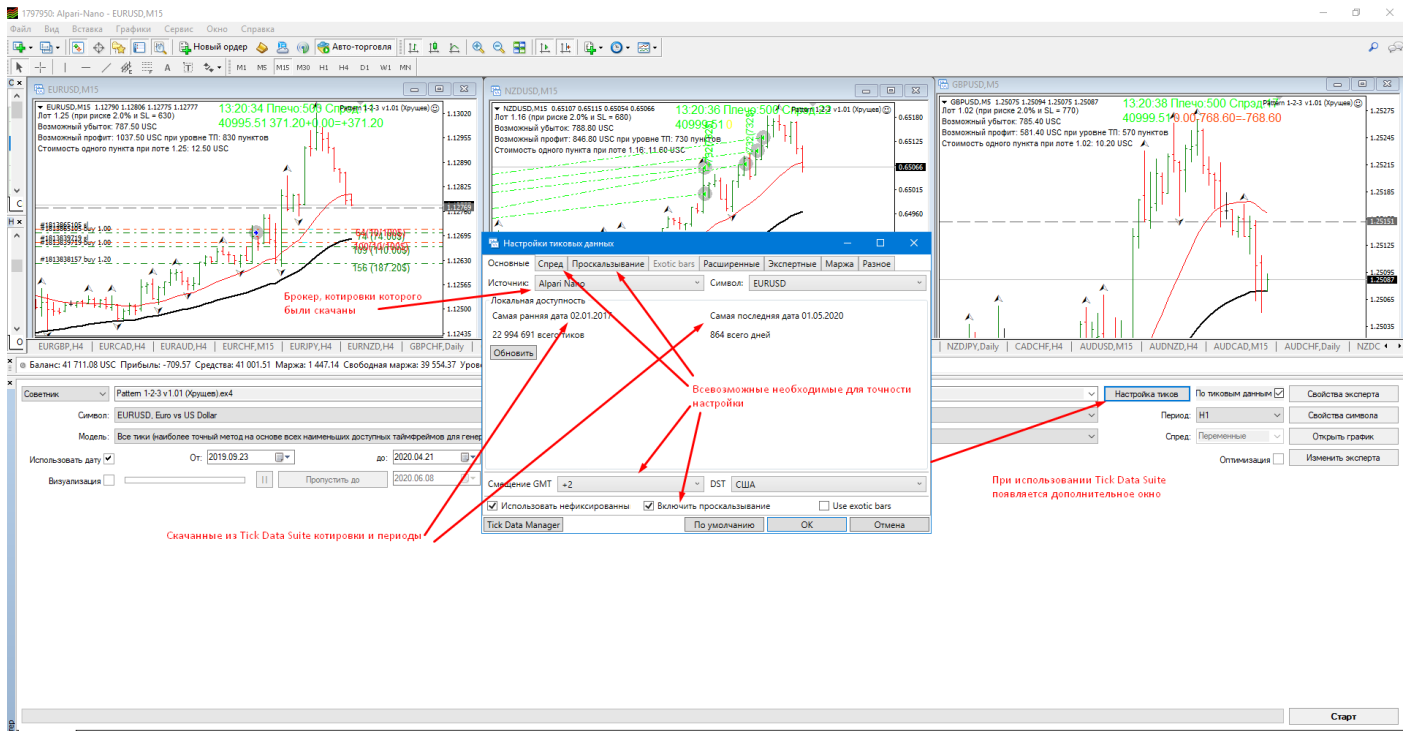
Как происходит оптимизация и тестирование роботов на данный момент в MT4

- 1) В MT4 через «Сервис / Архив котировок / Импорт» загружаются котировки в формате .csv или .txt. Одно движение на конкретно выбранном ТФ (на скринах это таймфрейм M15) включает в себя точную дату и время, цену открытия, цену закрытия, цену максимума, цену минимума и объём. Важный момент – в алгоритме MT4 бары (свечи) формируются от младшего таймфрейма (ТФ), поэтому критически важно найти котировки тиков для минутного ТФ (это не проблема). Далее на основании загруженных котировок ТФ M1 формируется ТФ M5, на основании ТФ M5 формируется M15 и так далее... Для более точного тестирования, я использую программу Tick Data Suite, из которой скачиваю котировки конкретного брокера, под которого тестирую роботы. Tick Data Suite автоматически привязывается к терминалу MT4 – в тестере MT4 появляется дополнительное меню для настроек тестирования. Результаты таких тестов намного лучше и правильнее.

Стандартная загрузка котировок



Меню Tick Data Suite



2) Далее открывается интерфейс встроенного тестера стратегий MT4. Выбирается необходимый алгоритм (торговый робот) для тестирования, торговый инструмент, выставляется дата, таймфрейм на котором будет проводиться оптимизация и тестирование и на котором в дальнейшем будет работать робот и т.д. (см. скрин)



Код советника (торгового робота) выглядит так

```

1 //-----
2 //
3 // Copyright 2005-2014, MetaQuotes Software Corp.
4 // http://www.mql4.com
5 //
6 #property copyright "2005-2014, MetaQuotes Software Corp."
7 #property link "http://www.mql4.com"
8
9 input double TakeProfit =50;
10 input double Lots =0.1;
11 input double TrailingStop =30;
12 input double MACDOpenLevel =9;
13 input double MACDCloseLevel =2;
14 input int MATrendPeriod =26;
15 //
16 //
17 //
18 void OnTick(void)
19 {
20     double MacdCurrent,MacdPrevious;
21     double SignalCurrent,SignalPrevious;
22     double MaCurrent,MaPrevious;
23     int cnt,sticks,total;
24 //---
25 // initial data checks
26 // it is important to make sure that the expert works with a normal
27 // chart and the user did not make any mistakes setting external
28 // variables (Lots, StopLoss, TakeProfit,
29 // TrailingStop) in our case, we check TakeProfit
30 // on a chart of less than 100 bars
31 //---
32 if(Bars<100)
33 {
34     Print("Bars less than 100");
35     return;
36 }
37 if(TakeProfit<10)
38 {
39     Print("TakeProfit less than 10");
40     return;
41 }
42 //--- to simplify the coding and speed up access data are put into internal variables
43 MacdCurrent=IMACD(NULL,0,12,26,9,PRICE_CLOSE,MODE_MAIN,0);
44 MacdPrevious=IMACD(NULL,0,12,26,9,PRICE_CLOSE,MODE_MAIN,1);
45 SignalCurrent=IMACD(NULL,0,12,26,9,PRICE_CLOSE,MODE_SIGNAL,0);
46 SignalPrevious=IMACD(NULL,0,12,26,9,PRICE_CLOSE,MODE_SIGNAL,1);
47 MaCurrent=IMA(NULL,0,MATrendPeriod,0,MODE_EMA,PRICE_CLOSE,0);
48 MaPrevious=IMA(NULL,0,MATrendPeriod,0,MODE_EMA,PRICE_CLOSE,1);
49
50 total=OrdersTotal();
51 if(total<1)
52 {
53     //--- no opened orders identified
54     if(AccountFreeMargin()<(1000*Lots))
55         Print("No have no money. Free Margin = ",AccountFreeMargin());
56 }
57 }
    
```

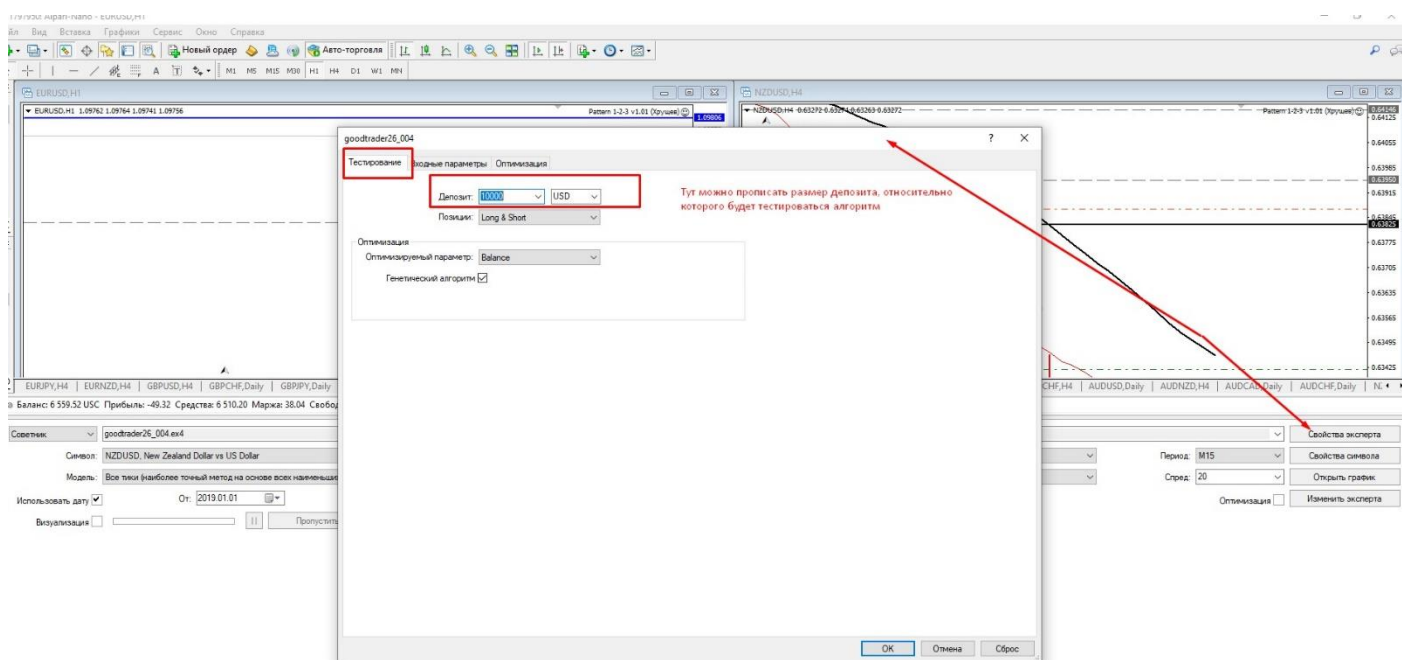
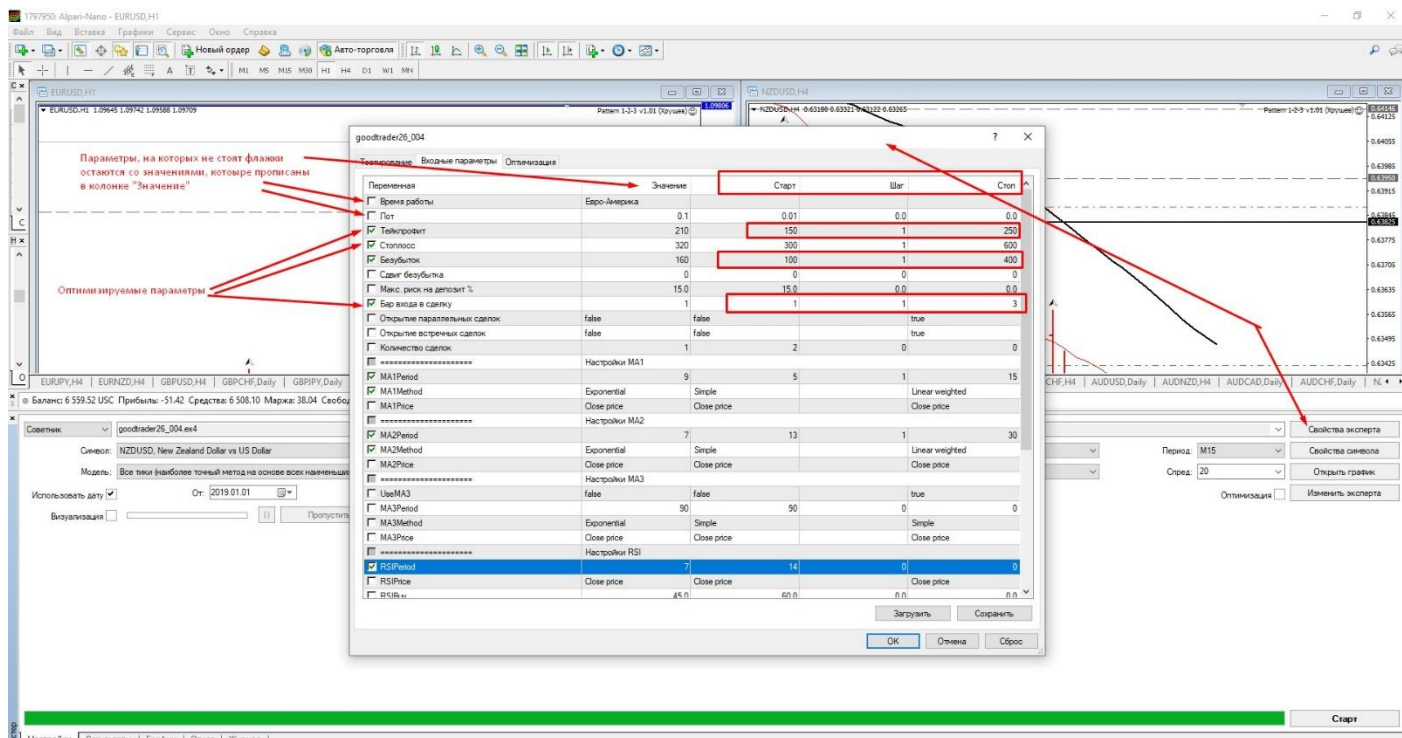
Для того, чтобы алгоритм работал и приносил прибыль на реальном счёте, он проходит два этапа тестирования. **Первый этап – оптимизация.** На этом этапе в настройках параметров на переменных, которые могут повлиять на конечный результат, ставят флажки и задают значения Старт / Шаг / Финиш. **Пример:** мы хотим понять, какой размер Стоп-Лосса и Тейк Профита в алгоритме даст максимально лучший прирост к депозиту. Мы задаем следующие значения:

Параметр	Стартовое значение	Шаг	Финишное значение
Стоп - Лосс	30	1	85
Тейк Профит	45	1	90

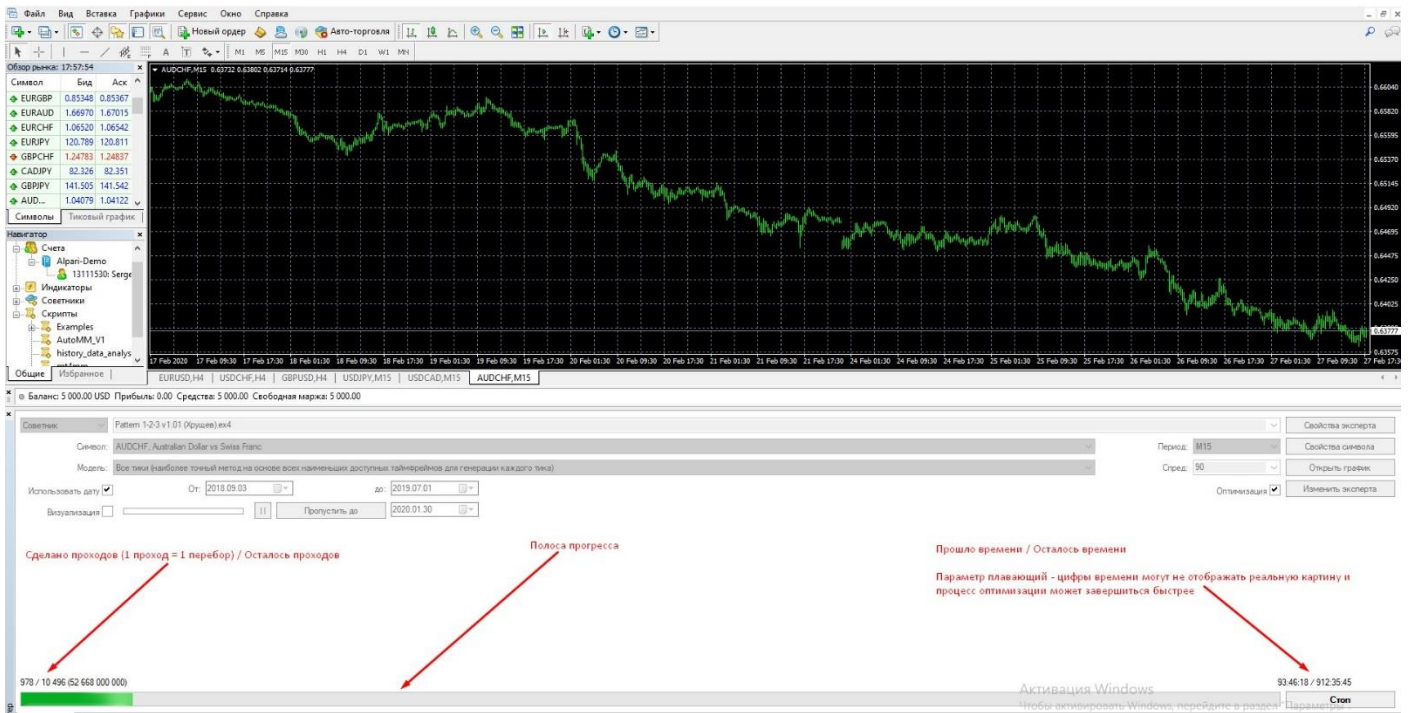
Во время оптимизации алгоритма, **методом перебора** будут выбраны некие значения из заданного диапазона, которые дадут лучший результат. Например, Стоп = 44 пункта, Тейк = 68 пунктов.

Точно также делается для всех остальных переменных на этапе оптимизации – ставятся флажки (это означает, что этот параметр будет оптимизироваться) и задаются значения Старт / Шаг / Финиш.

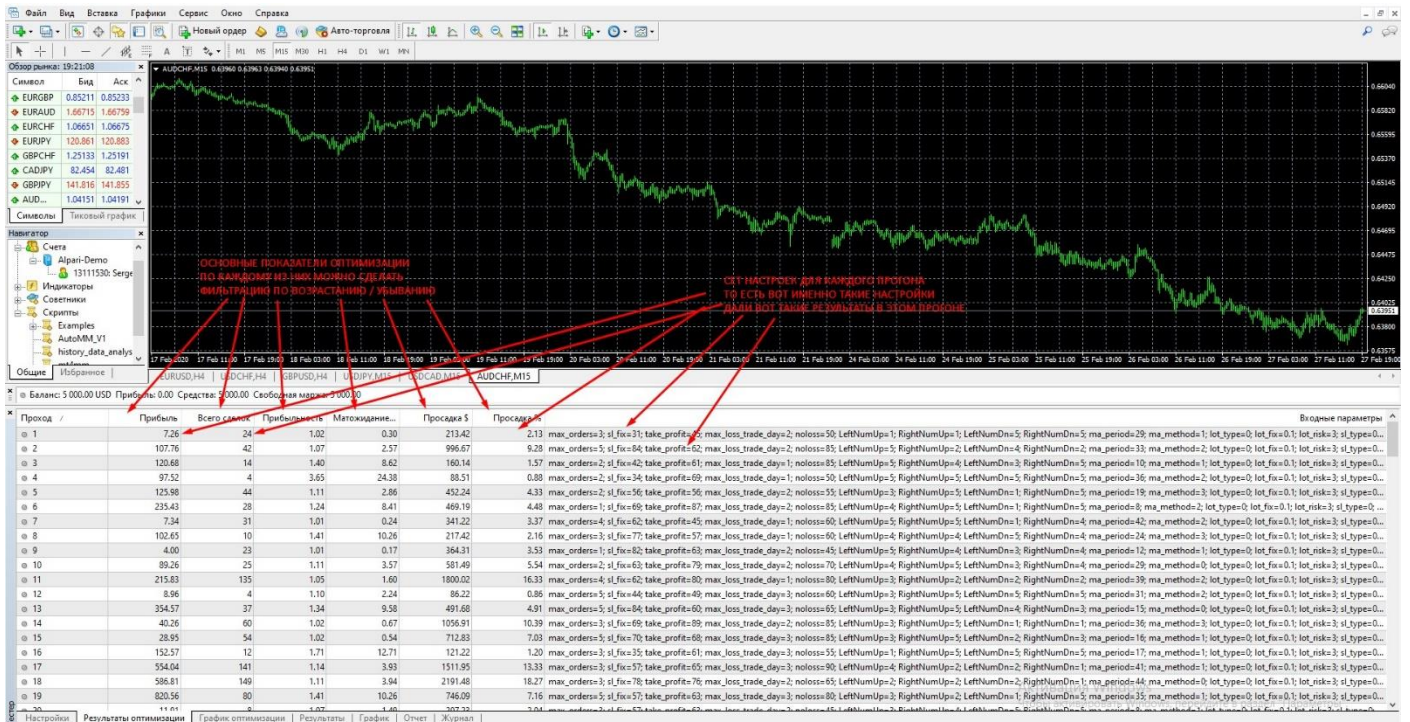
Окно «Свойства Эксперта»



Далее, после выставления всех параметров и установки флажка «Оптимизация» нажимается кнопка «Старт» и начинается процесс оптимизации. Выглядит он следующим образом:



Вот так выглядят результаты оптимизации:



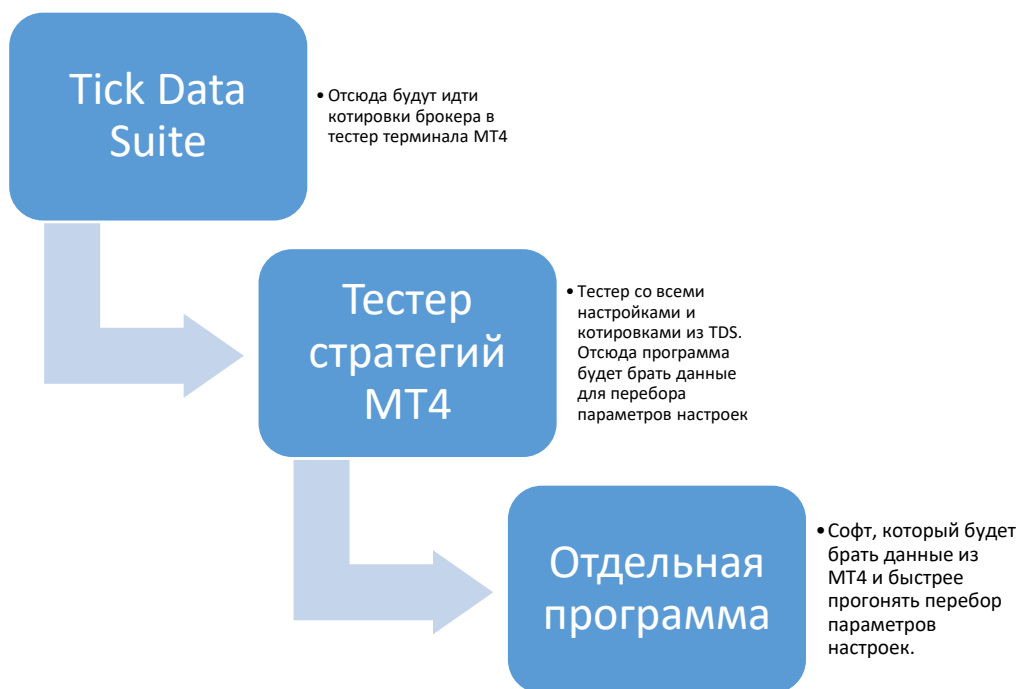
Временные параметры бэк и форвард теста: бэк-тест (оптимизация) проводится на периоде №1. Допустим мы хотим вывести настройки для работы робота сейчас (к примеру, у нас сегодня 01.02.2020). Период оптимизации мы выставляем 01.02.2018 – 01.02.2018 (2 года). На этом периоде робот как бы подбирает параметры, которые на рынке, который был в этот период, принесут максимальную прибыль (результаты оптимизации можно отфильтровать и увидеть настройки, которые показали максимальный результат). Затем в графах дат выставляется период 01.02.2018 – 01.02.2020 (2 года) – не обязательно именно такие даты и длина периода. И на этом периоде уже к роботу применяется настройка, выведенная оптимизацией и делается прогон на новом периоде и другом рынке. То есть на отрезке, на котором робот еще не работал и к нему не приспособлен. Суть такого тестирования в том, что на первом периоде робот подбирает параметры, а на втором доказывает, что эти же настройки, которые на прошлом периоде давали прибыль, на новом

периоде могут делать то же самое. Часто бывает ситуация, что те настройки, которые на периоде бэк-теста дают прибыль, на этапе форвард-теста сливает депозит. Задача – найти те настройки, которые на обоих периодах приносят прибыль.

Подробнее о оптимизации и форвард – тестировании можно почитать тут <https://avtoforex.ru/testirovanie/26-kak-optimizirovat-sovetnikov-v-metatrader-4.html>

Предположительное решение проблемы – отдельная программа.

- 1) Первый возможный вариант – сделать так, чтобы из МТ4 все данные транслировались в отдельную программу, в которой непосредственно и будет идти перебор параметров настроек и будут выдаваться результаты.



- 2) Второй вариант – писать полную копию тестера стратегий МТ4, но могут возникнуть проблемы с кодами роботов (роботы пишутся на языке MQL4 как раз под терминал МТ4) и скачиванием в новую программу котировок из Tick Data Suite (программа лицензионная платная и жестко привязана к МТ4).

Плюс в идеале, чтобы этот отдельный софт мог не только провести оптимизацию за временной период №1, но и автоматически форвард-тестом прогнать полученные результаты на периоде №2, а результаты, полученные на форварде (период №2) так же автоматически прогнать на следующем форвард-тесте за период №3. Итоговые результаты после периода №3 желательно должны быть в формате .set (Прибыльность 1,37 Матожидание 12,73 Просадка 14,65.set), который понимает МТ4.

Теперь на данном этапе мне необходим ваш ответ по возможности реализации какого-либо из вариантов, и я тогда по этому варианту уточню ТЗ по настройкам и прочему.

Резюме тезисно.

Сейчас весь процесс происходит следующим образом:

- 1) Из программы Tick Data Suite скачиваются котировки нужного брокера за определенный период. Данные доступны с 2013 года по сей день. Использование данных из TDS дает 99,9% точности моделирования работы робота при тестировании. То есть 99,9% того, что он так же будет работать и в реальной торговле. Плюс в настройках, которые даёт TDS, в MT4 можно настроить много переменных, которые влияют на качество тестирования и которые нельзя настроить в стандартном тестере стратегий MT4 без TDS (спред, проскальзывание, задержку исполнения ордеров, плечо и т.д.)
- 2) В тестере стратегий MT4 выбирается тестируемый робот, период, настраиваются все необходимые параметры.
- 3) Запускается период оптимизации на временном периоде №1. Цель – методом перебора подобрать параметры, при которых робот покажет наилучшие результаты торговли. Процесс оптимизации происходит автоматически без участия человека. Отнимает примерно 45% времени относительно всего процесса – с начала и до получения лучшего рабочего сета настроек.
- 4) После оптимизации получается список из нескольких тысяч результатов (сетов настроек) с показателями торговли каждого сета. Список отфильтровывается по параметру прибыль. Далее выбирается период бэк и форвард теста – период №2. Это период, на котором робот как бы еще не работал или работал частично. То есть начинается часть работы, которая исключает «подгонку» настроек робота под историю. В процессе бэк и форвард тестирования, КАЖДЫЙ сет кликается вручную человеком и прогоняется на выбранном участке времени. Полученные результаты (если они устраивают по показателям) также вручную сохраняются в папку терминала. Далек не каждый сет даст хорошие результаты на новом периоде. Процесс похож на вымывание золотого песка – среди 3-4 тысяч сетов найти хотя бы 30-50 сетов, которые покажут вменяемый результат. Данное тестирование занимает также 45% времени.
- 5) После периода бэк и форвард теста, начинается третий этап – форвард-тест 2. Берется временной период №3 и те результаты (сеты) которые мы сохраняли в папку терминала в предыдущем пункте, вручную загружаются из папки в тестер и прогоняются на периоде №3. Полученные результаты также сохраняются в папку терминала вручную. Лучший из этих результатов и будет непосредственно рабочим сетом настроек, который пойдёт на реальную торговлю. Отнимает 10% времени, т.к. отобранных сетов настроек уже не тысячи, а намного меньше – 20-30-50 +.

Весь процесс по нахождению рабочего сета для одной валютной пары (а этих пар примерно 20-25) занимает около 1-1,5 месяца – оптимизация и два тестирования на разных периодах. Плюс ко всему, рабочие настройки, которые попадают в реальную торговлю со временем устаревают (примерно за период 8-12 месяцев работы) и нуждаются в повторном процессе оптимизации и тестирования. Конечно, не на всех валютных парах робот может показывать приемлимые результаты по прибыльности и просадке, но даже если пар будет не 25, а 15, легче не становится. Проще говоря, чтобы один робот работал на тех же 15 валютных парах, нужны 15 сетов настроек, на поиск которых уйдет около года (можно запускать несколько процессов параллельно, но это от силы 3-4 процесса, но работа в параллели замедляет общий процесс). Следовательно, первые найденные настройки устареют и всё можно начинать снова по кругу.

Цель создания отдельного софта – автоматизировать пункты 3, 4 и 5, и ускорить процесс за счет легкости самой программы и за счет автоматизации ручных процессов – особенно в пункте 4. Понятно, что программа не будет делать необходимые расчеты и перебор параметров за минуту. Но

даже если нахождение рабочего сета будет происходить в автоматическом режиме скажем, даже за неделю – это намного лучше, чем 1 – 1,5 месяца и плюс много ручной работы.

Вторая часть ТЗ по Тестеру стратегий FOREX

Этот раздел включает в себя описание процесса перебора, модели тестирования, применение к нему фильтров, которые будут отсеивать заранее худшие сеты настроек, а также набросок интерфейса программы (не претендует на абсолют, но суть передаётся).

По факту генетический алгоритм (ГА), который применяется в встроенных тестерах Metatrader, ускоряет процесс оптимизации работы. Если в разрабатываемое нами ПО можно встроить ГА, или как-то интегрироваться с Metatrader – это один вопрос. Если не получится, то можно использовать и прямой математический перебор.

1) Процесс оптимизации и тестирования (идея)

ВАЖНО!!!! По возможности – добавить выбор одновременно нескольких торговых инструментов (3-5 валютных пар) и нескольких таймфреймов (2-3) для мультивалютного и мультитаймфреймного тестирования. Соответственно в итоговой таблице результатов для каждого сета настроек добавятся колонки торгового инструмента и таймфрейма.

Касаемо интеграции с Tick Data Suite – насчет переменных из меню настроек, они безусловно нужны (повторюсь, не прям все), а саму скачанную тиковую историю возможно будет подкачивать из того же файла, из которого её берет МТ4. То есть будут одновременно запущены и МТ4 и этот тестер. – просто мысли вслух.

Стандартный процесс нахождения рабочего сета настроек описан в первой части ТЗ. **Идея улучшения этого процесса состоит в следующем.**

Запускаем оптимизацию на периоде №1. При этом ограничиваем ее фильтрами (о них ниже). Происходит перебор параметров. Полученный сет настроек сразу уходит на второй этап, который идёт параллельно. А именно на бэк&форвард тест на периоде №2 (который также будет ограничиваться фильтрами) – результат второго прогона, если не удовлетворяет требованиям фильтрации, сразу отсеивается, если удовлетворяет – то этот сет настроек уходит на третий этап. Третий этап на периоде №3 – это чистый форвард тест, который также ограничивается фильтрами и уже его результаты будут финальными и будут выдаваться в итоговой таблице.

Весь процесс идёт параллельно! То есть проход (сет настроек) после оптимизации попадает сразу на второй этап, при этом оптимизация продолжает работать и искать новые сеты, а если сет настроек со второго этапа достоин попасть на третий – он отправляется туда и все эти процессы идут непрерывно до полного окончания тестирования.

Вопрос: насколько такая параллельность будет влиять на общую скорость, нежели те же самые процессы, но выполняемые поэтапно и отдельно друг от друга?

Для подтверждения работы с каждым из этапов, в интерфейсе ПО будет ставиться флажок. То есть может быть так, что прогон будет нужен только двухэтапный, тогда флажок с 3-го этапа (период №3) снимается. Количество выводов итоговых сетов в таблицу настраивается в настройке – то есть, это

2) Модели тестирования

Модель №1 – простая по периодам. В настройках и интерфейсе программы имеем три отдельных окна. Возле окна «Форвард Тест» должен быть флажок (использовать / не использовать) – возможно понадобится простой прогон оптимизация – тест.

Оптимизация		Бэк&Форвард Тест		Форвард Тест	
Период №1		Период №2		Период №3	
ДД:ММ:ГГ	ДД:ММ:ГГ	ДД:ММ:ГГ	ДД:ММ:ГГ	ДД:ММ:ГГ	ДД:ММ:ГГ

Модель №2 – Walking Forward Test. Схема модели указана в файле «Интерфейс программы». Смысл в проведении с определенным временным шагом нескольких прогонов «Оптимизация – Форвард Тест» с фиксацией контрольных результатов форвардного тестирования и их дальнейшем сравнении по завершении всего тестирования.

	2014				2015				2016				2017																							
ШАГ - ПОЛГОДА	ян14	фев14	мар14	апр14	мая14	июн14	июл14	авг14	сеп14	окт14	ноя14	дек14	ян15	фев15	мар15	апр15	мая15	июн15	июл15	авг15	сеп15	окт15	ноя15	дек15	ян16	фев16	мар16	апр16	мая16	июн16	июл16	авг16	сеп16	окт16	ноя16	дек16
	ОПТИМИЗАЦИЯ								ТЕСТИРОВАНИЕ																											
					ОПТИМИЗАЦИЯ								ТЕСТИРОВАНИЕ																							
									ОПТИМИЗАЦИЯ								ТЕСТИРОВАНИЕ																			
													ОПТИМИЗАЦИЯ								ТЕСТИРОВАНИЕ															
																	ОПТИМИЗАЦИЯ								ОПТИМ											

Процессинг Walking Forward Test происходит точно так же, как и в предыдущей модели. Допустим оптимизируем на периоде 2 года и форвард тест = полгода. Также ограничиваем и оптимизацию и форвард фильтрами (о них ниже) и начинается процесс оптимизации на периоде №1, попадаете подходящий сет настроек, и он сразу параллельно с процессом оптимизации идёт на форвард-тестирование. Отдельной настройкой нужно сделать, сколько результатов форварда будет сохраняться и выводиться в итоговую таблицу. И так прошли первый круг Оптимизация – Форвард, сохранили X результатов первого форварда – начинается второй такой же круг. И так далее вплоть до полного окончания. Все результаты форвардов сводятся в итоговую таблицу (с возможностью выгрузки таблицы в Эксель)

Итоговая таблица результатов Walking Forward Test

№	Начальная дата	Конечная дата	Чистая прибыль, \$	Прибыльность (Profit-Factor)	Матожидание	Макс. Просадка, %	Кол-во сделок	Кол-во Лонг сделок	Кол-во Шорт сделок	Кол-во убыточных сделок, %	Фактор Восстановления (Recovery Factor)	Прибыль в пунктах (средн. За месяц)	R - Квадрат (описан в разделе Фильтры)
1	янв.16	июн.16											
2	янв.16	июн.16											
3	янв.16	июн.16											
4	янв.16	июн.16											
5	янв.16	июн.16											
6	июл.16	дек.16											
7	июл.16	дек.16											
8	июл.16	дек.16											
9	июл.16	дек.16											
10	июл.16	дек.16											
11	дек.16	июн.17											
12	дек.16	июн.17											
13	дек.16	июн.17											
14	дек.16	июн.17											
15	дек.16	июн.17											
16	июн.17	дек.17											
17	июн.17	дек.17											
18	июн.17	дек.17											
19	июн.17	дек.17											
20	июн.17	дек.17											

3) Фильтры

- **Количество сделок (не менее) –**
- **Количество Шорт сделок (не менее X% от общего количества сделок) –**
- **Количество Лонг сделок (не менее X% от общего количества сделок) –** Эти две настройки нужны для того, чтобы отсеивать подгонку. То есть если из, скажем, 500 сделок 100 шорт, а 400 лонг – на лицо явная подгонка параметров.
- **Количество убыточных сделок (не более, %) –**
- **Фактор восстановления (Recovery Factor) -** = Чистая прибыль / Максимальную просадка (в валюте депозита). Должен быть не менее 5 за 1 год, а в идеале 15-20. Этими цифрами и будет регулироваться этот фильтр.
- **Прибыль в пунктах (четырёхзнак) за месяц по одному торговому инструменту –**
- **Просадка (не более, %) –** считается по формуле (Макс. Просадка в \$ / Начальный депозит) * 100%
- **Матожидание прибыли (не менее, указывается простое число (5-10-12-25 и т.д.) –**
- **Прибыльность (Profit Factor) –**
- **Коэффициент R – квадрат** (этот параметр необязательно должен быть фильтром, но по возможности его хорошо бы добавить в итоговый отчет для цифрового анализа кривой баланса, чтобы не заморачиваться с её визуализацией) – подробно описан тут <https://www.mql5.com/ru/articles/2358#c10>
- **Итоговый баланс (не менее, \$) –** тут указывается сумма в валюте депозита, ниже которой начальный баланс не должен опускаться при фиксации всех сделок. Допустим у нас стартовый депозит 10 000\$, в данном параметре указываем к примеру 8 000\$. Следовательно, если в процессе какого-либо прогона настроек, баланс пересекает черту в 8 000\$, данный прогон отсеивается.
- **Непрерывное количество убыточных сделок (не более) –** 2-3-4-5 и т.д. = простое число

Фильтры должны применяться к каждой оптимизации и тестированию, независимо от этапа и модели тестирования, для того, чтобы отсеивать заведомо худшие и не подходящие под критерии показателей нормальной модели сетевых настроек, тем самым ускоряя весь процесс оптимизации / тестирования. К оптимизации допустим можно применять более мягкую фильтрацию, чтобы совсем уж не сужать круг возможных сетевых настроек, а вот к форвард-тестированию можно применить фильтры и жестче.

ВАЖНО!!!! Всё, что измеряется в пунктах – в программе и в алгоритмах указывается в пунктах (!!!) по четырёхзнаку. Не в пипсах по пятизнаку!