

УДОСКОНАЛЕНА МЕТОДИКА ПОБУДОВИ АВТОМАТИЗОВАНОЇ ТОРГОВОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ПАТЕРНІВ «PRICE ACTION»

Л.І. Мещеряков, С.І. Випанасенко, Н.С.Дрешпак, Д.Г. Клімов
(Україна, Дніпро, НТУ «Дніпровська політехніка»)

В процесі глобалізації та установки тісних економічних відносин між різними країнами, в першу чергу між США та Європою, появи міжнародних компаній і корпорацій, виникає і світовий фінансовий ринок, як величезна торгова площадка для продавців і покупців.

Фінансовий ринок – являє собою систему відносин, що виникає в процесі обміну економічних благ з використанням грошей в якості активу-посередника.

На фінансовому ринку відбувається мобілізація капіталу, надання кредиту, здійснення обмінних грошових операцій і розміщення фінансових коштів у виробництві. А сукупність попиту та пропозиції на капітал кредиторів і позичальників різних країн утворює світовий фінансовий ринок.

Світовий фінансовий ринок – це частина світового ринку позичкового капіталу, сукупність попиту і пропозиції на капітал кредиторів і позичальників різних країн. Одним із сегментів світового фінансового ринку є фондовий ринок або ринок цінних паперів.

В останнє десятиріччя на світових біржах брокерів-людей багато в чому потіснили так звані «торгові роботи» – комп'ютерні програми, спеціалізовані для світової біржової торгівлі.

Торговий робот – це автоматизована торгова система, яка частково або повністю замінює людину під час роботи на біржі, при цьому робот може керуватися трейдером (наприклад, прийняття рішень про відкриття/закриття позиції приймає сам трейдер) або працювати за заздалегідь складеною оперативною програмою.

Роботи, які самостійно ведуть торги на біржі – це спеціально розроблені програми. Вони будуються на математичних алгоритмах, і можуть відстежувати показники різних індексів на фондовій біржі та на основі отриманих даних здійснювати операції з купівлі або продажу. Звичайний обсяг угод автоматизованих систем в кілька разів перевищує кількість угод, які здійснюють звичайні люди.

Фахівці зазвичай ділять біржових роботів на три групи: трендові (дирекціоні або спрямовані), контртрендові та арбітражні. Вони відповідають різним типам торгових стратегій.

Звичайне завдання трендового робота – якнайшвидше вловити тенденцію зростання або падіння котирувань та відкрити позицію. Після чого своєчасно

«відчути» розворот тренда і встигнути зафіксувати прибуток (тобто продати акції або валюту). Контртрендові роботи намагаються «зловити» всі відкати ціни, особливо вони добре працюють у флетовому стані ринку. У свою чергу, арбітражний робот повинен отримувати прибуток, виявляючи перекося в цінах на ідентичні або тісно пов'язані активи на різних ринках.

Алгоритмічна торгівля, або Алгоритмічний трейдинг – це метод виконання великої заявки (занадто великої, щоб бути виконаною за один раз). За допомогою особливих алгоритмічних інструкцій така заявка (parentorder) ділиться на кілька під-заявок (childorders) зі своїми характеристиками ціни та обсягу і кожна з під-заявок відправляється в певний час на ринок для виконання. Такі алгоритми були придумані для того, щоб трейдерам не доводилося постійно стежити за котируваннями і ділити велику заявку на менші вручну. Популярні алгоритми носять назви "PercentageofVolume", "Pegged", "VWAP", "TWAP", "ImplementationShortfall", "TargetClose".

При цьому ліквідність фінансових інструментів зазвичай оцінюють за обсягом і кількістю здійснюваних операцій (обсяг торгів), величиною спреда між кращими цінами попиту та пропозиції (максимальними цінами заявок на покупку і мінімальними цінами заявок на продаж) і сумарного обсягу заявок поблизу кращих цін попиту і пропозиції (ціни та обсяг поточних заявок можна побачити в склянці торгового терміналу). Чим більше обсяг і кількість угод по інструменту, тим більше його торгова ліквідність, в свою чергу, чим менше різниця між кращими цінами попиту та пропозиції і чим більше обсяг заявок поблизу цих цін, тим більше моментальна ліквідність.

З появою ринку та інвестицій було розроблено велику кількість методик прогнозування руху ціни найрізноманітніших торгових інструментів: від акцій і валютних пар, до сировинних товарів, ф'ючерсів та індексів. Всім цим можна торгувати заради отримання прибутку або хеджування ризиків вже відкритих ринкових позицій. Кожен тип ринку (валютний, фондовий, ф'ючерсний тощо) має свої особливості ціноутворення, а тому одні і ті ж види аналізу можуть бути по-різному ефективні.

Історично так склалося, що всі види аналізу можна зарахувати по двом основним категоріям – технічний і фундаментальний аналіз. В їх основі лежить дві абсолютно різні характеристики, які відображає поточна ринкова ситуація. Технічний аналіз передбачає роботу з історичними даними, що формують графік руху котирувань. Фундаментальний аналіз передбачає роботу з економічними показниками, які допоможуть знайти недооцінені активи. Завдання по розробці торгового робота ставить питання про те, чи можна використовувати фундаментальний аналіз і його методики при програмуванні механізмів торгівлі, адже це невід'ємний тип прогнозування руху активів і ціни. Фундаментальний аналіз – термін для позначення методів прогнозування ринкової (біржової) вартості торгового інструменту, заснованих на аналізі фінансових і виробничих показників, що характеризують його потенційний прибуток. Фундаментальний

аналіз використовується інвесторами для оцінки вартості компанії (або її акцій), яка відображає стан справ в компанії, рентабельність її діяльності. При цьому аналізі піддаються вивченню фінансові показники компанії: виручка, EBITDA (Earnings Before Interests, Taxes, Depreciation and Amortization), чистий прибуток, чиста вартість компанії, зобов'язання, грошовий потік, величина виплачуваних дивідендів і виробничі показники компанії.

Інструментарій реалізації здійснено на мові програмування MQL 4 – яка є процедурною Сі-подібною мовою, що містить спеціалізовані можливості, які спрощують написання прикладних програм, і виконуються з-під торгового терміналу, для автоматизації торговельної діяльності. Вона містить стандартні для Сі типи даних і має можливість написання призначених для користувача функцій. Всі програми мають вбудований джерело даних у вигляді чисельного уявлення графіка зміни ціни того активу, для якого дана програма була запущена. Для того, щоб зрозуміти можливості торгової системи, розглянемо схему взаємодії програми на MQL 4 з ринком (рис. 1).

В клієнтському терміналі є інформаційне середовище – набір параметрів, які інформують про стан ринку і відносини між трейдером і дилінговим центром. Воно містить відомості про поточні ціни, обмеження щодо мінімального і максимального розміру ордера, мінімальної дистанції стоп-наказів, дозвіл / заборону здійснювати автоматичну торгівлю і безліч інших корисних параметрів, що характеризують поточну ситуацію. Інформаційне середовище оновлюється з надходженням в термінал нових тиків.



Рис. 1. Программа на MQL4 в составе клиентского терминалу MetaTrader 4

Існує проблема неоднозначної інтерпретації сигналів, коли один і той же індикатор може генерувати різні сигнали в одній певній ринкової ситуації. Звідси з'являється наступна проблема – необхідно не просто виявляти сигнали з різних джерел, а грамотно їх розпізнавати, щоб в результаті можна було визначити: це точка входу на покупку або на продаж. Саме тому основою для виявлення точки входу в ринок є утворення патерна Price Action і вже в момент його утворення використовуються сигнали інших індикаторів в якості підтвердження або спростування. Кожен сигнал тут представлений у вигляді простого масиву з двох елементів, в якому міститься задана вага сигналу і тип (на продаж або на покупку). Сигнали на продаж містять значення «- 1», відповідно, сигнали на покупку містять значення «+1». При відкритті торгової позиції значення типу всіх виявлених сигналів складаються. Якщо сума більша нуля, відкривається позиція на покупку, якщо менше – на продаж. У доповнення до цього, при підрахунку вагових коефіцієнтів, вага сигналів, які мають відмінний від точки входу тип сигналу, при обчисленні сили даної точки, віднімається, а не додається. В налаштуваннях можна включити опцію «Ігнорувати протилежні сигнали» (`ignore_enemy_signals`), тоді суперечливі сигнали не враховуватимуться.

Відомо, що в залежності від різних умов, в яких проводиться оцінка графіка руху ціни, ефективність одних і тих же інструментів аналізу сильно відрізняється.

Крім того, було доведено, що одні торговельні сигнали можуть підкріплюватися іншими – наприклад, відскік ціни від рівня опору збігається з сигналом перетину ліній індикатора MACD в зоні перекупленості. В такому випадку, MACD є фільтром, який підтверджує або спростовує початковий торговий сигнал, а саме підхід і відскік ціни від лінії опору. Звідси стає зрозуміло, що для більш точного входу в ринок потрібно грамотно розробити алгоритмічну систему, яка не просто буде визначати різні цінові рівні, тренди і виявляти сигнали. Алгоритм повинен містити механізм, який дозволить комбінувати кілька сигналів в один, фільтрувати суперечливі і, на основі заданих параметрів, виміряти «силу» точки входу, утворену патерном Price Action. Для цього був використаний механізм вагових коефіцієнтів. Визначення вагового коефіцієнта наступне: це числовий коефіцієнт, параметр, що відображає значимість, відносну важливість, «вагу» даного чинника, показника в порівнянні з іншими чинниками, що впливають на досліджуваний процес.

З точки зору поставленої задачі, кожен торговий сигнал від кожного індикатора, об'єкта чи патерна Price Action має свою «вагу», в залежності від таймфрейма. Під час аналізу підраховується вага всіх торгових сигналів, які були виявлені, потім множаться на коефіцієнт активного таймфрейма і на коефіцієнт пройденої відстані. Формалізовано це можна представити таким виразом:

$$S_s = \sum VS_i * \left(1 + \frac{D}{w/10}\right) * cT,$$

де $\sum VS_i$ – сума вагових коефіцієнтів;

D – діапазон пошуку сигналу (за замовчуванням 1000);

w – відстань від поточної ціни до ціни закриття свічки, яка утворила потенційну точку входу;

cT – коефіцієнт таймфрейма (для H1 дорівнює 2, H4 – 3, D1 – 4).

Вагові коефіцієнти для патернів Price Action обчислюються в діапазоні від 1 до 5, для інших джерел сигналів від 1 до 2,5. Для відкриття позиції необхідно, щоб сила точки входу подолала певний числовий поріг (за замовчуванням 700).

Реалізація компоненту аналізу торгового інструменту містить в собі набір попередньо інтегрованих і призначених для користувача функцій щодо здійснення пошуку точок входу в ринок і подальшого аналізу, в разі знаходження такої точки. Відомо, що під точкою входу мається на увазі торговий сигнал від утворення патерну Price Action.

Пошук патернів Price Action здійснюється від попередньої свічки в заданому в налаштуваннях діапазоні. Для визначення кожного простого патерну, що складається з однієї свічки, розроблена функція, яка порівнює параметри свічки відповідно до параметрів патерну. Кожна функція має тип bool і повертає true у випадку, якщо патерн був знайдений (рис. 2).



Рис. 2. Результат пошуку патернів Price Action і точки входу, які вони генерують, на графіку валютної пари NZD/USD H1

З рис. 2 видно, що виявлених патернів досить багато, саме тому за замовчуванням рекомендується використовувати тільки найбільш сильні патерни, наприклад такі як «Пін-бар» і його різновиди, «Поглинання» або «Розворотнемарібозу». Проте в алгоритмі реалізовано визначення 38 патернів Price Action, щоб користувач мав можливість вибору з широкого спектру різних торгових комбінацій. Компонент аналізу торгового інструменту містить в собі набір попередньо інтегрованих і призначених для користувача функцій щодо здійснення пошуку точок входу в ринок і подальшого аналізу, в разі знаходження такої точки. Під точкою входу мається на увазі торговий сигнал від утворення патерну Price Action.

Слід зазначити, що такі інструменти аналізу, як опціонний аналіз, побудова рівнів корекції Фібоначі та використання цих рівнів в якості визначення важливих цінових зон відкритого інтересу, не були використані. Також не використовувались більшість із патернів PriceAction, тому що при тестуванні було визначено, що чим більше патернів використовується, тим більше помилкових сигналів генерується, а це впливає на якість роботи та стабільність торгового рахунку, тому були обрані лише три. Підводячи підсумки, треба відмітити той факт, що автоматизована торгова система з використанням

патернів Price Action, як основного джерела аналізу, тестувалася на реальному рахунку з втручанням трейдера і показала до початкового депозиту позитивну динаміку.

На основі аналізу тестування розробленої автоматизованої торгової системи в режимі реального часу на деморахунку та на реальному, і в повністю автономному режимі та с втручанням трейдера стає можливим зробити висновок, що розроблений торговий робот є ефективним і може приносити стабільний прибуток.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Adamatzky A. (Ed.) *Advances in Unconventional Computing. Volume 2. Prototypes, Models and Algorithms.* – Springer International Publishing, 2017. – 812 p.
2. Albers V., Still B. (Eds.) *Usability of Complex Information Systems: Evaluation of User Interaction.* – CRC Press, 2011. – 392 p.
3. Arlow Jim, Neustadt Ila. *EnterprisePatterns and MDA: Building Better Software with ArchetypePatterns and UML.* – Addison–Wesley Professional, 2004. 528 p.
4. Aslaksen E.W. *Designing Complex Systems: Foundations of Design in the Functional Domain.* – Auerbach Publications; 1 edition (October 27, 2008). – 176 p.
5. Barrera D.G., Diaz M. *Communicating Systems with UML 2: Modeling and Analysis of Network Protocols.* – ISTE Ltd. – Wiley, 2011. – 268 p.
6. Baskerville R., De Marco M., Spagnoletti P. (Eds.) *Designing Organizational Systems: An Interdisciplinary Discourse.* – Springer, 2013 – 342 p.
7. Boehm Barry W. *A Spiral Model of Software Development and Enhancement.* – Object Management Group, 2010. – 180 p.
8. Borland. *Delphi 7: Developer's Guide.* Author and Publishers: Borland Software Corporation, 2002. – 1108 p.
9. Braun A.W. *Enterprise Software Delivery.* – Addison–Wesley, 2012. – 291 p.
10. Camp O., Filipe J.B.L., Hammoudi S., Piattini M. *Enterprise Information Systems V.* – Kluwer, 2005. – 339 p.
11. Cantù M. *Delphi 2010 Handbook: A Guide to the New Features of Delphi 2010.* – CreateSpace Independent Publishing Platform, 2010. – 318 p.
12. Cantu M. *Mastering Delphi 7.* – Sybex, 2003. – 992 p.