

УДК 681.324:519.17

## ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ ПРИ ПАРАЛЛЕЛЬНОЙ ОПТИМИЗАЦИИ ФИНАНСОВЫХ СТРАТЕГИЙ НА КЛАСТЕРНОЙ СИСТЕМЕ

О.Г. Монахов

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск

E-mail: monakhov@rav.sccc.ru

Описан подход к оптимизации финансовых стратегий (алгоритмов), основанный на индикаторах финансовых и товарных рынков и эволюционных вычислениях. Представлен параллельный генетический алгоритм, который был применен для автоматизации поиска оптимальных параметров торговых стратегий с точки зрения максимизации показателей доходности. Экспериментально с использованием кластерной системы и GPU-ускорителей показано, что предложенный алгоритм позволяет увеличить доходность финансовых стратегий и имеет отличную масштабируемость и ускорение при параллельных вычислениях на суперЭВМ с 53000 ядер.

### Ключевые слова:

Параллельные вычисления, генетический алгоритм, финансовые стратегии, оптимизация, графические процессоры, биржевая торговля.

### Key words:

Parallel computation, genetic algorithm, financial strategies, optimization, graphics processing units, trading.

### Введение и постановка задачи

В практике биржевой торговли одним из основных направлений при выработке финансовых стратегий (торговых алгоритмов) является технический анализ ценовых рядов с помощью множества индикаторов [1–3]. В соответствии с принятой торговой стратегией, выраженной в виде набора правил, с поведением ценового ряда и значениями индикаторов инвестор принимает решение о совершении/не совершении сделки купли-продажи в данный момент времени. При совершении сделки инвестор руководствуется соображениями максимизации доходности и минимизации риска. Принятый набор правил, составляющий торговую стратегию, и используемые индикаторы имеют эмпирический характер, и значения их параметров определяются, в основном, опытным путем (методом проб и ошибок). Однако, как показывают эксперименты, такой подход с использованием известных правил и статически задаваемых параметров часто приводит к убыточным стратегиям. Использование высокопроизводительных вычислительных систем для торговли на бирже обозначается термином «высокочастотная» алгоритмическая торговля (high-frequency algorithmic trading) и позволяет компьютерным программам (торговым роботам) самостоятельно отслеживать данные по нескольким индексам на фондовых биржах, оптимизировать торговые стратегии и совершать миллионы сделок за максимально короткий промежуток времени.

В работе рассматривается проблема поиска параметров данной стратегии биржевой торговли  $S$  с целью оптимизации заданной целевой функции  $F$ , характеризующей ее качество. Будем считать, что цена на акцию представлена в виде ценового ряда  $\{C_i\}$ ,  $1 \leq i \leq N$ , с заданной частотой (например, минутные или часовые цены), где  $C_i$  – цена закрытия в момент  $i$ . Пусть  $r_{i+1} = C_{i+1} - C_i$ . Важными инстру-

ментами технического анализа рынка акций являются скользящие средние, индикаторы и осцилляторы, на основе которых формируются множество торговых стратегий и которые помогают инвестору принимать решения о купле-продаже акций [1–3].

Пусть мы имеем индикатор технического анализа:  $I_i^{(n)} = f(C_i, C_{i-1}, \dots, C_{i-n})$ . Обобщенная торговая стратегия  $S(I_i^{(n)})$ , основанная на индикаторе  $I_i^{(n)}$ , определяется следующими соотношениями:

$$\varphi_{i+1} = \begin{cases} 1, & \text{если } I_i^{(n)} > \varepsilon_1, \\ \varphi_i, & \text{если } -\varepsilon_2 \leq I_i^{(n)} \leq \varepsilon_1, \\ -1, & \text{если } I_i^{(n)} < -\varepsilon_2. \end{cases}$$

где  $\varepsilon_1, \varepsilon_2 > 0$  – уровни значимого изменения индикатора  $I_i^{(n)}$ . Состояние покупки в данной торговой стратегии наступает при  $\varphi_{i+1} = 1$ , а состояние продажи наступает при  $\varphi_{i+1} = -1$ . Решение о сделке (купли/продаже) принимается при смене состояний:  $\varphi_i \varphi_{i+1} = -1$ .

Эта стратегия  $S(I_i^{(n)})$  будет использована как темплейт (с некоторыми модификациями) для определения торговых стратегий на основе различных индикаторов технического анализа, и поиска оптимальных значений свободных параметров  $(n, \varepsilon_1, \varepsilon_2)$ , определяющих стратегию с наилучшими показателями доходности, будет осуществляться с помощью генетического алгоритма (ГА).

Например, одним из часто используемых индикаторов при анализе ценовых рядов является экспоненциальное скользящее среднее порядка  $k$ :

$$\bar{C}_{i+1}^{(k)} = \bar{C}_i^{(k)} + \frac{2}{k+1}(C_{i+1} - \bar{C}_i^{(k)}); \bar{C}_0^{(k)} = C_0, \quad 1 \leq i \leq N.$$

Порядок скользящего среднего  $k$  определяет степень сглаживания цены: чем больше  $k$ , тем сильнее сглаживание. Рассчитывается также разность экспоненциальных скользящих средних порядков  $k_1 < k_2$ :  $r_{-i} = (\bar{C}_i^{(k_1)} - \bar{C}_i^{(k_2)}) / \bar{C}_i^{(k_2)}$ .

Приведем пример простейшей торговой стратегии на основе экспоненциальных скользящих средних [2]. Задается уровень значимого изменения сглаженных цен  $\varepsilon > 0$ . Состояние покупки в данной торговой стратегии наступает при  $\bar{r}_i > \varepsilon$ , а состояние продажи наступает при  $\bar{r}_i < -\varepsilon$ . Решение о сделке (куплю/продажу) принимается при смене состояний. Стратегия имеет три свободных параметра  $k_1, k_2, \varepsilon$ , изменение которых изменяет показатели доходности и риска торговой стратегии.

Поиск оптимальных стратегий (с наилучшими показателями доходности и/или риска) может осуществляться для каждого типа акций отдельно в динамике торговых сессий, с постоянной адаптацией к рыночной ситуации, или в квазидинамическом режиме, когда расчет оптимальных параметров происходит либо через заданные периоды времени, либо по выполнению определенных условий (например, по достижении заданного уровня потерь).

Пусть торговая стратегия  $S$  содержит параметры  $P = \{p_n\}$ ,  $n > 0$ , описывающие значения целочисленных и действительных коэффициентов и переменных, значения индексов, параметры структур данных, константы и некоторые примитивные операции алгоритма (величины инкрементов и декрементов, знаки переменных, логические операции и отношения, типы округления переменных).

Целевая функция  $F$  оценивает величину доходности стратегии  $S$ , полученную при заданных значениях параметров  $P = \{p_n\}$ ,  $n > 0$  и при входных данных ценового ряда  $\{C_{ij}: F_i = F_i(P, C_j), j \leq i, 1 \leq i \leq N\}$ .

Таким образом, проблема оптимизации торговой стратегии состоит в следующем: для данной стратегии  $S$  и заданного набора значений ценового ряда  $\{C_{ij}, 1 \leq i \leq N\}$ , необходимо найти такие значения параметров  $P^*$  стратегии  $S$ , что  $F_N(S(P^*, C_{ij})) \geq F_N(S(P, C_{ij}))$ , для  $1 \leq i \leq N$ , при любых других значениях параметров  $P \in \text{Dom}(P)$ .

Целью данной работы является описание подхода к оптимизации торговых стратегий, основанного на эволюционных вычислениях, и его распараллеливанию на кластерной суперЭВМ с графическими (GPU – graphics processing unit) ускорителями. Представлен генетический алгоритм, который в процессе торговых сессий осуществляет автоматический поиск оптимальных параметров торговых стратегий и индикаторов с точки зрения максимизации показателей доходности. Для решения данной проблемы в работе предлагается подход, основанный на применении генетических алгоритмов [4, 5] с использованием предварительного знания прикладной области (множества индикаторов), выборе обобщенной схемы торговой стратегии, задаваемой в виде темплейта с параметрами [6], и ограничении пространства поиска оптимальных параметров.

#### Генетический алгоритм оптимизации стратегий

Генетический алгоритм основан на моделировании процесса естественного отбора в популяции особей, каждая из которых представлена точкой

в пространстве решений задачи оптимизации. Особи представлены структурами данных  $Gen$  – хромосомами, включающими свободные (неопределенные) параметры  $p_k$  торговой стратегии  $S$ :  $Gen = \{P\} = \{p_1, p_2, \dots, p_k\}$ ,  $k > 0$ . Эти параметры определяют необходимую торговую стратегию  $S(Gen)$ . Каждая популяция является множеством структур данных  $Gen$  и определяет множество стратегий  $S(Gen)$ .

Примем, что целевая функция (fitness function, функция качества, функция пригодности)  $F$  вычисляет суммарную доходность  $D_N$ , полученную в результате торговли в соответствии с данной стратегией  $S$  за  $N$  шагов для заданного ценового ряда  $\{C_{ij}, 1 \leq i \leq N\}$ :

$$F = D_N = \sum_{m=1}^{N_{br}} (d_m^{br} - Comm),$$

где  $d_m^{br} = \frac{C_m^{sell} - C_m^{buy}}{C_m^{buy}}$ ;  $C_m^{sell}$ ,  $C_m^{buy}$  – цены продажи и

покупки в  $m$ -той сделке;  $N_{br}$  – число сделок за  $N$  шагов моделирования;  $Comm$  – размер постоянных комиссионных за каждую сделку. Целью алгоритма является поиск максимума  $F$ .

Основная идея генетического алгоритма состоит в эволюционном преобразовании множества хромосом (параметров стратегии) в процессе естественного отбора с целью выживания «сильнейшего». В нашем случае этими особями являются стратегии, имеющие наибольшее значение целевой функции. Алгоритм начинается с генерации начальной популяции. Все особи в этой популяции создаются случайно, затем отбираются наилучшие особи и запоминаются. Для создания популяции следующего поколения (следующей итерации) новые особи формируются с помощью генетических операторов селекции (отбора), мутации, кроссовера и добавления новых элементов (для сохранения разнообразия популяции).

Оператор *мутации* применяется к особям, случайно выбранным из текущей популяции с вероятностью  $p_{mut} \in [0, 1]$ . Мутация хромосомы  $Gen$  состоит в изменении значения случайно выбранного параметра  $p_k$  на другую, случайно выбранную величину из множества допустимых значений. Оператор *кроссовера (скрещивания)* применяется к двум особям (родителям), случайно выбранным из текущей популяции с вероятностью  $p_{cross} \in [0, 1]$ . Кроссовер состоит в порождении двух новых особей путем обмена частями хромосом родителей. Оператор *создания нового элемента* (особи) состоит в генерации случайных значений параметров  $p_k$ . Это позволяет увеличить степень разнообразия особей при создании популяции. Оператор *селекции (отбора)* реализует принцип выживания наиболее приспособленных особей. Он выбирает наилучших особей с минимальными значениями целевой функции  $F$ .

Для поиска оптимума заданной целевой функции  $F$  итерационный процесс вычислений в генетическом алгоритме организован следующим образом.

*Первая итерация:* порождение начальной популяции. Все особи популяции создаются с помощью оператора *новый элемент*, с проверкой и отсеиванием всех непригодных особей. После заполнения массива популяции лучшие особи отбираются и запоминаются в массиве *best*. *Промежуточная итерация:* шаг от текущей к следующей популяции. Основной шаг алгоритма состоит в создании нового поколения особей на основе массива *best*, используя операции селекции, мутации, кроссовера и добавления новых элементов. После оценки целевой функции для каждой особи в поколении проводится сравнение величин этих функций с величинами целевых функций тех особей, которые сохранены в массиве *best*. В том случае, если элемент из нового поколения лучше, чем элемент *best* [*i*], для некоторого *i* помещаем новый элемент на место *i* в массив *best* и сдвигаем в нем все остальные элементы на единицу вниз. Таким образом, лучшие элементы локализуются в верхней части массива *best*. *Последняя итерация (критерий останова):* итерации заканчиваются либо после исполнения заданного числа шагов, либо после нахождения оптимального значения целевой функции *F*.

#### Экспериментальные результаты

Предложенный генетический алгоритм с использованием темплейтов был успешно применен для адаптивной оптимизации торговых стратегий, основанных на следующих, наиболее популярных инструментах технического анализа: экспоненциальных скользящих средних (EMA – exponential moving average), индекса относительной силы (RSI – relative strength index), темпа изменения цены (ROC – price rate-of-change), момента (Momentum), метода схождения/расхождения скользящих средних (MACD – Moving Average Convergence/Divergence) [1–3].

Для экспериментов были рассмотрены ценовые ряды с минутными интервалами для акций ГАЗПРОМа (10000 точек), РАО ЕС России (10000 точек), NIKKEI (10000 точек), DJIA – Dow Jones Industrial Average (10000 точек), для периода с 16.04.2006 по 16.06.2006. Мы использовали первые 5000 точек для обучения и остальные точки – для тестирования.

Число итераций и размер популяции выбирались экспериментальным путем, основываясь на параметрах из [4, 5]. Значения основных параметров в экспериментах следующие: размер популяции – от 524288 до  $105 \cdot 524288$ , число итераций – 100, частота мутации – 0,15, частота кроссовера – 0,7, комиссионные – 0,001.

Генетический алгоритм оптимизации торговых стратегий был реализован в системе эволюционно-синтеза алгоритмов на основе шаблонов (TES – template-based evolutionary synthesis) [6] на языке программирования C. Параллельная реализация ГА оптимизации стратегий биржевой торговли выполнена на основе распараллеливания по данным

[7–9], при этом, в отличие от аналогичных подходов [10], на графических процессорах исполнялись все части генетического алгоритма (и генетические операторы, и вычисление целевой функции), а не отдельные его элементы, что позволило на порядок увеличить ускорение. Так, результаты сравнения параллельной (на видеокарте NVIDIA GeForce 470 GTX 1280MB, 448 процессоров) и последовательной (на одном ядре процессора INTEL Core2Quad Q6700, 2.66 ГГц) реализации ГА для оптимизации стратегии показали ускорение времени выполнения алгоритма в 178 раз.

Параллельная реализация генетического алгоритма оптимизации стратегий биржевой торговли была выполнена в Сибирском Суперкомпьютерном Центре на кластерной суперЭВМ НКС-30Т с гибридной архитектурой (содержащей 40 вычислительных модулей, каждый из которых содержит 2 CPU Intel Xeon X5670 (по 6 ядер) и 3 графических ускорителя Tesla M 2090 (по 512 ядер), всего 480 ядер CPU и 61440 ядер GPU). Программа реализована в системе программирования CUDA [11] с использованием библиотеки MPI путем распараллеливания по данным с равномерным распределением популяции по потокам графической подсистемы. На каждом вычислительном модуле использовались 3 ядра CPU (3 потока MPI, по одному на каждую карту, для организации коммуникаций) и все 1536 ядер GPU – для вычислений. В конце итераций среди всех потоков выбирается лучшее решение, что минимизирует взаимодействия и позволяет получить отличное масштабирование и значительное (линейное) ускорение для параллельного алгоритма. Величина популяции составляла 524288 особей на каждой видеокарте (графическом ускорителе), использовались ценовые ряды в 10000 точек. Отметим, что в случае реализации генетического алгоритма на ГПУ, данные для обучения и тестирования следует по возможности помещать или в быструю разделяемую память или в константную память. Так, размещение этих данных в константной памяти, которую могут использовать все потоки сразу, позволило сократить время исполнения алгоритма на ГПУ почти в два раза.

В табл. 1 приведены результаты сравнения для параллельной реализации генетического алгоритма для оптимизации стратегии с MACD для акций ГАЗПРОМа при числе вычислительных модулей  $M$ , числе графических ускорителей  $N$ , числе ядер GPU  $K$ , размере популяции  $Pop$ , времени исполнения  $T$  (сек.) и полученном ускорении  $S_p$  по отношению к одному процессору GPU. Из табл. 1 видно, что при линейном возрастании объема вычислений (при линейном росте величины популяции) время исполнения остается постоянным (с отклонениями не более 1%), что свидетельствует об отличном масштабировании, высокой эффективности распараллеливания и линейном ускорении (превышающим два порядка) для параллельного генетического алгоритма.

**Таблица 1.** Сравнение параллельных реализаций генетического алгоритма

| M  | N   | K     | Pop        | T (сек) | Sp  |
|----|-----|-------|------------|---------|-----|
| 1  | 1   | 512   | 524288     | 785,45  | 1   |
| 1  | 3   | 1536  | 3*524288   | 779,4   | 3   |
| 5  | 15  | 7680  | 15*524288  | 781,24  | 15  |
| 10 | 30  | 15360 | 30*524288  | 784,17  | 30  |
| 15 | 45  | 23040 | 45*524288  | 787,78  | 45  |
| 20 | 60  | 30720 | 60*524288  | 785,42  | 60  |
| 25 | 75  | 38400 | 75*524288  | 784,96  | 75  |
| 30 | 90  | 46080 | 90*524288  | 784,47  | 90  |
| 35 | 105 | 53760 | 105*524288 | 784,62  | 105 |

**Таблица 2.** Увеличение суммарной доходности торговых стратегий

| Тип акции \ Индикатор | EMA | MACD | RSI | ROC  |
|-----------------------|-----|------|-----|------|
|                       | %   |      |     |      |
| ГАЗПРОМ               | 155 | 14,3 | 75  | 67,5 |
| DJA                   | 41  | 35,7 | 31  | 16,1 |

#### СПИСОК ЛИТЕРАТУРЫ

1. Achelis S.B. Technical analysis from A to Z. – Chicago: Probus, 1996. – 380 p.
2. Артемьев С.С., Якунин М.А. Математическое и статистическое моделирование на фондовых рынках. – Новосибирск: ИВМиМГ СО РАН, 2003. – 123 с.
3. LeBeau Ch.L., David W. Computer analysis of the futures market. – New-York: IRWIN, 1992. – 323 p.
4. Goldberg D.E. Genetic Algorithms in Search, Optimization and Machine Learning. – MA: Addison-Wesley, 1989. – 432 p.
5. Koza J. Genetic Programming. – Cambridge: The MIT Press, 1992. – 609 p.
6. Монахов О.Г. Эволюционный синтез алгоритмов на основе шаблонов // Автометрия. – 2006. – № 1. – С. 106–116.
7. Монахов О.Г., Монахова Э.А. Параллельные системы с распределенной памятью: структуры и организация взаимодействий. – Новосибирск: Изд-во СО РАН, 2000. – 242 с.
8. Монахов О.Г. Параллельный генетический алгоритм для оптимизации торговых стратегий // Сиб. журн. вычисл. математики – 2008. – № 4. – С. 423–432.
9. Монахов О.Г. Параллельные генетические алгоритмы при поиске оптимальных финансовых стратегий на графических процессорах // Параллельные вычисления и задачи управления (РАСО-2010): Труды IV Междунар. конф. – Москва, 2010. – С. 212–224.
10. Tsutsui S., Fijimoto N. Solving Quadratic Assignment Problems by Genetic Algorithms with GPU Computation: A Case Study // In Proc. of Internat. Genetic and Evolutionary Computation Conference (GECCO-2009). – Montreal Quebec, Canada, 2009. – P. 176–181.
11. NVIDIA CUDA Programming Guide. URL: <http://developer.nvidia.com/cuda/cuda-downloads> (дата обращения 10.08.2012).

Поступила 17.09.2012 г.