УДК 681.324+519.17

ОПТИМИЗАЦИЯ ТОРГОВЫХ СТРАТЕГИЙ С ПОМОЩЬЮ ПАРАЛЛЕЛЬНЫХ ЭВОЛЮЦИОННЫХ ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ

О. Г. Монахов[1]

Описан подход для оптимизации торговых стратегий (алгоритмов), основанный на индикаторах финансовых и товарных рынков и эволюционных вычислениях. Представлен параллельный генетический алгоритм, реализованный на графических процессорах NVIDIA в технологии CUDA для автоматизации поиска оптимальных параметров торговых стратегий с точки зрения максимизации показателей доходности.

Ключевые слова: торговые стратегии, параллельный генетический алгоритм, финансовый индикатор, CUDA, эволюционные вычисления.

1. Введение и постановка задачи. В практике биржевой торговли одним из основных направлений при выработке торговых стратегий (торговых алгоритмов) является технический анализ ценовых рядов с помощью множества индикаторов [1–6].

В соответствии с принятой торговой стратегией, выраженной в виде набора правил, с поведением ценового ряда и значениями индикаторов инвестор принимает решение о совершении/несовершении 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, а 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 мощных 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.

Целью настоящей статьи является описание подхода к оптимизации торговых стратегий, основанного на эволюционных вычислениях. Представлен параллельный генетический алгоритм (ГА), реализованный на графических процессорах NVIDIA в технологии CUDA, который в процессе торговых сессий осуществляет автоматический поиск оптимальных параметров торговых стратегий и индикаторов с точки зрения максимизации показателей доходности.

Считаем, что цена на акцию представлена в виде ценового ряда $\{C_i\}$, $1 \leqslant i \leqslant N$, с заданной частотой τ (например, минутные или часовые цены), C_i — цена закрытия в момент i. Пусть $r_{i+1} = C_{i+1} - C_i$.

Важными инструментами технического анализа рынка акций являются скользящие средние, индикаторы и осцилляторы, на основе которых формируется множество торговых стратегий и которые помогают инвестору принимать решения о купле-продаже акций [1–6].

Пусть мы имеем индикатор технического анализа: $I_i^{(n)} = f(C_i, C_{i-1}, \ldots, C_{i-n})$.

Пусть ε_1, $\varepsilon_2 > 0$ — уровни значимого изменения индикатора $I_i^{(n)}$. Тогда обобщенная торговая стратегия $S\big(I_i^{(n)}\big)$, основанная на индикаторе $I_i^{(n)}$, будет определяться следующими соотношениями:

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

Состояние покупки в данной торговой стратегии наступает при $\varphi_{i+1} = 1$, а состояние продажи — при $\varphi_{i+1} = -1$. Решение о сделке (купле/продаже) принимается при смене состояний: $\varphi_i \varphi_{i+1} = -1$.

[1] Институт вычислительной математики и математической геофизики СО РАН, просп. акад. Лаврентьева, 6, 630090, Новосибирск; вед. науч. сотр., e-mail: monakhov@rav.sscc.ru

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

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

$$\overline{C}_{i+1}^{(k)} = \overline{C}_i^{(k)} + \frac{2}{k+1}\Big(C_{i+1} - \overline{C}_i^{(k)}\Big), \quad 0 \leqslant i \leqslant N-1, \quad \overline{C}_0^{(k)} = C_0.$$

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

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 $\overline{r}_i > \varepsilon$, а состояние продажи наступает при $\overline{r}_i < -\varepsilon$. Решение о сделке (купле/продаже) принимается при смене состояний. Стратегия имеет три свободных параметра k_1, k_2 и ε, 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). Частота работы алгоритма оптимизации параметров стратегий при биржевой торговле зависит от динамики рынка и горизонта инвестирования, особенно часто это происходит при работе в течение дня на малых тайм-фреймах (минутных и тиковых интервалах), когда оценка и адаптация параметров стратегий происходит в реальном времени через 100–200 интервалов.

Пусть торговая стратегия S содержит параметры $P = \{p_n\}$, $n \geqslant 0$, 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 $P = \{p_n\}$ и при входных данных ценового ряда C_i: $F_i = F_i\big(S(P, C_j)\big)$, $j \leqslant i$, $1 \leqslant i \leqslant N$.

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

Для решения данной проблемы в настоящей статье предлагается подход, основанный на применении параллельной версии [10] генетических алгоритмов (ГА) [7, 8] с использованием предварительного знания прикладной области (множества индикаторов), выборе обобщенной схемы торговой стратегии, задаваемой в виде темплейта с параметрами [9], и ограничении пространства поиска оптимальных параметров.

2. Генетический алгоритм. 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 в себя свободные (неопределенные) параметры p_k торговой стратегии S: $\mathrm{Gen} = \{P\} = \{p_1, p_2, \ldots, p_k\}$, $k \geqslant 0$. Эти параметры определяют необходимую торговую стратегию $S(\mathrm{Gen})$. Каждая популяция является множеством структур данных Gen и определяет множество стратегии $S(\mathrm{Gen})$.

Основная идея алгоритма синтеза 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 операций селекции (отбора), мутации, кроссовера и добавления новых элементов (для сохранения разнообразия популяции).

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 $\{C_i\}$, $1 \leqslant i \leqslant N$: $F = D_N = \sum\limits_{m=1}^{N_{\mathrm{br}}} \big(d_m^{\mathrm{br}} - \mathrm{Comm}\big)$, где

$d_m^{\mathrm{br}} = \dfrac{C_m^{\mathrm{sell}} - C_m^{\mathrm{buy}}}{C_m^{\mathrm{buy}}}$, C_m^{sell}, C_m^{buy} — цены продажи и покупки в m-й сделке, N_{br} — число сделок за N шагов моделирования, Comm — размер постоянных комиссионных за каждую сделку. Целью алгоритма является поиск максимума F.

3. Представление данных. Основными структурами данных в программной реализации эволюционного алгоритма являются хромосомы Gen. Для представления и реализации хромосомы была предложена линейная структура для параметров p_k. Линейная структура хромосомы Gen используется для представления различных типов параметров p_k стратегии, таких как значения целочисленных и действительных коэффициентов и переменных, значения индексов, величины инкрементов и декрементов, а также знаков переменных, логических операций и отношений, типов округления переменных [9, 10].

Приведем пример линейной структуры хромосомы, каждый ген которой обозначен через [g] и соответствует одному из параметров стратегии: $\{[\ 5\ ]\,[1.2]\,[-1]\,[+]\,[\&]\,[\geqslant]\,[\lfloor x \rfloor]\}$.

При создании хромосом Gen задаются значения параметров p_k, по которым можно производить оценивание и модификации стратегии $S(\mathrm{Gen})$.

Таким образом, для фиксированных значений параметров p_k мы можем вычислять значения целевой функции F на основе заданных стратегий $S(\mathrm{Gen})$ и полученных в ходе эволюции хромосом Gen для требуемого ценового ряда $\{C_i\}$, $1 \leqslant i \leqslant N$. После выполнения стратегий S для данного ценового ряда мы получаем значения целевой функции F и выбираем лучшие стратегии в популяции.

4. Операторы алгоритма. Оператор *мутации* применяется к особям, случайно выбранным из текущей популяции с вероятностью $p_m \in [0, 1]$. Мутация линейной хромосомы Gen состоит в замене 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 $p_c \in [0, 1]$. Кроссовер состоит в порождении двух новых особей путем обмена частями хромосом родителей (обмен подчастями линейных структур хромосомы).

Оператор создания *нового элемента* (особи) состоит в генерации случайных значений параметров p_k.

Оператор *селекции* (отбора) реализует принцип выживания наиболее приспособленных особей. Он выбирает наилучших особей с минимальными значениями целевой функции.

Заметим, что только простейшие генетические операторы были использованы в алгоритме, но данный подход позволяет применять и более сложные операторы, разработанные для ГА [7, 8].

5. Итерационный процесс. Для поиска оптимума заданной целевой функции F итерационный процесс вычислений в ГА 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. Ключевой шаг алгоритма состоит в создании нового поколения особей на основе массива best и текущей популяции при помощи операций 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. Если элемент из нового поколения лучше, чем элемент best[i] для некоторого i, то 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 остановки): итерации заканчиваются либо после исполнения заданного числа шагов, либо после нахождения оптимальной стратегии $S(\mathrm{Gen})$ (с заданным значением целевой функции F).

После выполнения заданного количества шагов алгоритма мы получаем множество (популяцию) стратегий $S(\mathrm{Gen})$, содержащее в элементе best[0] стратегию $S^*(\mathrm{Gen})$, имеющую максимальное значение целевой функции F.

6. Распараллеливание генетического алгоритма и 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) [1–6].

Для экспериментов были рассмотрены ценовые ряды с минутными интервалами для акций ГАЗПРОМа (тип 1, 10 000 точек) и РАО ЕЭС России (тип 2, 10 000 точек) за период с 16.04.2006 по 16.06.2006. Первые 5000 точек были использованы для обучения, а остальные точки — для тестирования.

Число итераций и размер популяции выбирались экспериментальным путем, основываясь на параметрах из [7, 8]. Значения основных параметров в экспериментах приведены в табл. 1.

Таблица 1
Значения параметров

Параметр	Значение
Размер популяции	98 304–1 548 288
Число итераций	30–5000
Частота кроссовера	70%
Частота мутации	15%

Генетический алгоритм оптимизации торговых стратегий был реализован в системе эволюционного синтеза алгоритмов на основе шаблонов (TES — Template-based Evolutionary Synthesis) [9] на языке программирования C. В первой серии экспериментов параллельная реализация генетического алгоритма оптимизации стратегий биржевой торговли была выполнена для графических процессоров G92 на видеокартах NVIDIA GeForce 8800 GT 512MB (112 процессоров) и GeForce 8800 GTS 512MB (128 процессоров) в системе программирования CUDA [11] с использованием библиотеки OpenMP при одновременном выполнении задания на двух видеокартах. В этом случае распараллеливание по данным сводится к равномерному распределению популяции по потокам системы. В конце итераций среди всех потоков выбирается лучшее решение. Величина популяции составляла 98 304 на каждой карте. В связи с ограничением на память, использовались ценовые ряды в 10 000 точек. Эксперименты показали, что при данных параметрах и ограничениях время выполнения задания для оптимизации стратегии на системе из двух указанных видеокарт эквивалентно времени выполнения на кластерной системе НКС-160 (Новосибирский Кластерный Суперкомпьютер) для 160 процессоров Itanium 2, 1.6 ГГц, т.е. для рассмотренной задачи эти две системы (кластерная и графическая) показывают приблизительно равную производительность.

Во второй серии экспериментов параллельная реализация генетического алгоритма оптимизации стратегий биржевой торговли выполнена также на основе распараллеливания по данным [10] для графического процессора Fermi GF100 на видеокарте NVIDIA GeForce 470 GTX 1280MB (448 процессоров, 1215 МГц) в системе программирования CUDA [11]. И в этой серии использовались ценовые ряды в 10 000 точек, и в конце итераций среди всех потоков выбирается лучшее решение, что минимизирует взаимодействия и позволяет получить значительное ускорение (до 178). 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 в два раза. Отметим также, что, как показали эксперименты на ГПУ, унификация потоков и сокращение различий между ними (что достигается путем уменьшения числа условных операторов в программе) позволяет уменьшить время исполнения генетического алгоритма на 30%.

Таблица 2
Сравнение параллельной и последовательной реализаций генетического алгоритма

N	Pop	T (сек) INTEL Q6700	T (сек) NVIDIA 470 GTX	Sp
1	229 376	21 090	166	127
2	516 096	62 080	363	171
3	1 032 192	128 800	723	178
4	1 548 288	131 900	1081	122

В табл. 2 приведены результаты сравнения параллельной (на видеокарте NVIDIA GeForce 470 GTX 1280MB, 448 процессоров) и последовательной (на одном ядре процессора INTEL Core2Quad Q6700, 2.66 ГГц) реализаций генетического алгоритма для оптимизации стратегии с MACD для акций ГАЗПРОМа при размере популяции Pop, времени исполнения T (сек.) и полученном ускорении Sp по отношению к процессору INTEL Q6700. При этом в ЦПУ-реализации векторизация циклов и распараллеливание на несколько ядер ЦПУ не осуществлялись. Генетические алгоритмы, работающие на ГПУ и ЦПУ, выполняли примерно одинаковые арифметические операции и давали приблизительно (с разницей не более 1%) одинаковые результаты. Эксперименты показали, что при указанных параметрах и ограничениях время выполнения задания для оптимизации стратегии на видеокарте NVIDIA GeForce 470 GTX эквивалентно времени выполнения на кластерной системе с 122–178 процессорными ядрами с частотой 2.66 ГГц.

Используемый генетический алгоритм позволил найти значения параметров торговых стратегий, обеспечивающие увеличение функции суммарной доходности (на 14%–167% для различных индикаторов, см. табл. 3) по сравнению с известными ранее [1, 3, 6].

7. Заключение. Представленный подход к оптимизации торговых стратегий, основанный на индикаторах технического анализа, эволюционных вычислениях и темплейтах, был успешно применен для поиска свободных параметров стратегий с целью максимизации функции суммарной доходности.

Параллельная реализация генетического алгоритма оптимизации стратегий биржевой торговли на графических процессорах NVIDIA позволяет получить существенное ускорение, превышающее два порядка, по сравнению с одноядерной реализацией на процессорах INTEL и сравнимое с линейным ускорением, полученным на кластерной системе НКС-160, а также увеличить значения функции суммарной доходности до 167%.

Таблица 3

Увеличение суммарной доходности торговых стратегий

Тип акции	Индикатор			
	EMA	MACD	RSI	ROC
ГАЗПРОМ	105%	14.3%	75%	67.5%
РАО ЕС России	167%	15.7%	151%	14.1%

Дальнейшее развитие рассмотренного подхода будет направлено на эволюционный синтез [9] новых торговых алгоритмов, правил и стратегий с использованием комбинаций нескольких индикаторов, поиском новых функций для анализа ценовых рядов и исследованием подходов для распараллеливания на кластеры, в узлах которых используются ГПУ.

СПИСОК ЛИТЕРАТУРЫ

1. *Achelis S.B.* Technical analysis from A to Z. Chicago: Probus, 1996.
2. *Артемьев С.С., Якунин М.А.* Математическое и статистическое моделирование на фондовых рынках. Новосибирск: ИВМиМГ СО РАН, 2003.
3. *LeBeau Ch., Lucas D.W.* Computer analysis of the futures market. New-York: IRWIN, 1992.
4. *Weissman R.L.* Mechanical trading systems. Hoboken: Wiley, 2005.
5. *Salov V.* Modeling maximum trading profits with C++ : new trading and money management concepts. Hoboken: Wiley, 2007.
6. *Blau W.* Momentum, direction and divergence. Hoboken: Wiley, 2001.
7. *Goldberg D.E.* Genetic algorithms in search, optimization and machine learning. Reading: Addison-Wesley, 1989.
8. *Koza J.* Genetic programming. Cambridge: MIT Press, 1992.
9. *Монахов О.Г.* Эволюционный синтез алгоритмов на основе шаблонов // Автометрия. 2006. № 1. 106–116.
10. *Монахов О.Г., Монахова Э.А.* Параллельные системы с распределенной памятью: структуры и организация взаимодействий. Новосибирск: Изд-во СО РАН, 2000.
11. *Боресков А.В., Харламов А.А.* Основы работы с технологией CUDA. М.: ДМК Пресс, 2010.

Поступила в редакцию
15.10.2011