

ПАРАЛЛЕЛЬНЫЕ ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ ПРИ ПОИСКЕ ОПТИМАЛЬНЫХ ФИНАНСОВЫХ СТРАТЕГИЙ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ ¹

О.Г. Монахов

Институт вычислительной математики и математической геофизики СО РАН

Россия, 630090, Новосибирск, Лаврентьева пр., 6

E-mail: monakhov@rav.sccc.ru

Ключевые слова: торговые стратегии, параллельный генетический алгоритм, технический анализ, графический процессор, темплейт, эволюционные вычисления

Key word: trading strategies, optimization, parallel genetic algorithm, Graphics Processing Unit, GPU, evolutionary computation

Описан подход для оптимизации торговых 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 на графических процессорах NVIDIA.

Parallel genetic algorithms for search optimal trading strategies on GPU / O.G. Monakhov (Institute of Computational Mathematics and Mathematical Geophysics SB RAS, 6 Pr. Lavrentieva, Novosibirsk, 630090, Russia, E-mail: monakhov@rav.sccc.ru), An approach for optimization of trading strategies (algorithms) based on indicators of financial markets and evolutionary computation is described. Parallel version of a genetic algorithm for search of optimal parameters of trading strategies for maximization of trading profit on GPU from NVIDIA is presented.

1 Введение и постановка задачи

Использование высокопроизводительных вычислительных систем в режиме "высокочастотной" торговли (high-frequency trading) на бирже позволяет компьютерным программам (торговым роботам) самостоятельно отслеживать данные по нескольким индексам на фондовых биржах и совершать миллионы сделок за максимально короткий промежуток времени. Торговые роботы начинают активно продавать или покупать ценные бумаги в зависимости от кратковременного изменения цены. За счет своих скоростей такие участники рынка способны зарабатывать больше, чем "классические" трейдеры, и оказывать заметное влияние на рынок. Таким образом, использование высокопроизводительных вычислительных систем и параллельных алгоритмов при разработке и реализации торговых стратегий является одной из актуальных задач финансовой математики и практики 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/не совершении сделки купли-продажи в данный момент времени.

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

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

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект N 08-01-00857)

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:

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

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 поиск оптимальных значений свободных параметров (n, ε) , определяющих стратегию с наилучшими показателями доходности, будет осуществляться с помощью ГА. 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 $P = \{p_n\}$, $n \geq 0$, 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 $C_i : F_i = F_i(S(P, C_j))$, $j \leq i$, $1 \leq i \leq N$.

Таким образом, проблема оптимизации торговой стратегии состоит в следующем: для данной стратегии S и заданного набора значений ценового ряда C_i , $1 \leq i \leq N$, необходимо найти такие значения параметров P^* стратегии S , что

$$F_N(S(P^*, C_i)) \geq F_N(S(P, C_i))$$

для $1 \leq i \leq N$, при любых других значениях параметров $P \in \text{Dom}(P)$.

Для решения данной проблемы в работе 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 Основные структуры данных и 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 свободные (неопределенные) параметры p_k торговой стратегии S : $Gen = \{P\} = \{p_1, p_2, \dots, p_k\}$, $k \geq 0$. Эти параметры определяют необходимую торговую стратегию $S(Gen)$. Каждая популяция является множеством структур данных Gen и определяет множество стратегии $S(Gen)$.

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 $\{C_i\}$, $1 \leq i \leq N$:

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

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 .

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].

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

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

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. Основной шаг алгоритма состоит в создании нового поколения особей на основе массива *best* и текущей популяции, 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 локализируются в верхней части массива *best*.

Последняя итерация (критерий остановки): итерации заканчиваются либо после исполнения заданного числа шагов, либо после нахождения оптимальной стратегии $S(Gen)$ (с задан-

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

3 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, 10000 точек) и РАО ЕЭС России (тип 2, 10000 точек), для периода 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	98304
Число итераций	500
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. Параллельная реализация генетического алгоритма оптимизации стратегий биржевой торговли была выполнена для графических процессорах G92 на видеокартах NVIDIA (1) GeForce 8800 GT 512MB (112 процессоров) и (2) GeForce 8800 GTS 512MB (128 процессоров) в системе программирования CUDA [11] с использованием 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 98304 на каждой карте. В связи с ограничением на память, использовались ценовые ряды в 10000 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 для 160 процессоров Itanium 2, т.е. для данной задачи эти две системы показывают приблизительно равную производительность.

Используемый генетический алгоритм позволил найти значения параметров торговых стратегий, обеспечивающие увеличение функции суммарной доходности (на 14% - 167% для различных индикаторов, см. Табл.2) по сравнению с известными ранее [1],[3],[6].

Таблица 2: Увеличение суммарной доходности торговых стратегий

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

4 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, сравнимое с линейным ускорением, полученным на кластерной системе НКС-160, и увеличить значения функции суммарной доходности.

Дальнейшее развитие данного 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 новых подходов для распараллеливания.

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. 158 с.
- [3] LeBeau, Charles and Lucas, David W. Computer analysis of the futures market, New-York: IRWIN, 1992.
- [4] Weissman, Richard L. Mechanical Trading Systems, Hoboken, New Jersey: John Wiley and Sons, Inc., 2005.
- [5] Salov, Valerii, Modeling maximum trading profits with C++ : new trading and money management concepts, Hoboken, New Jersey: John Wiley and Sons, Inc., 2007.
- [6] Blau, W. Momentum, Direction and Divergence, Hoboken, New Jersey: John Wiley and Sons, Inc., 2001.
- [7] Goldberg, D. E. Genetic Algorithms, in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley, 1989.
- [8] Koza, J. Genetic Programming. Cambridge: The MIT Press, 1992.
- [9] Монахов О. Г. Эволюционный синтез алгоритмов на основе шаблонов // Автометрия. 2006. N1. С.106-116.
- [10] Монахов О.Г., Монахова Э.А. Параллельные системы с распределенной памятью: структуры и организация взаимодействий, Новосибирск: Изд-во СО РАН, 2000. 242 с.
- [11] NVIDIA CUDA Programming Guide. Available at [http : //developer.nvidia.com/object/gpucomputing.html](http://developer.nvidia.com/object/gpucomputing.html)

