

Реализация генетического алгоритма оптимизации финансовых стратегий на кластерной системе и графических процессорах

О.Г. Монахов

*Институт вычислительной математики и математической геофизики СОРАН
Новосибирск*

Введение и постановка задачи

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 инструментами 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 [1-6].

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 модификациями) для определения торговых стратегий на основе различных индикаторов технического анализа, и поиск оптимальных значений свободных параметров $(n, \varepsilon_1, \varepsilon_2)$, 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 :

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

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

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 (купли/продаже) принимается при смене состояний. Стратегия имеет три свободных параметра k_1, k_2, 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).

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 $\{C_i\}$: $F_i = F_i(S(P, C_j)), j \leq i, 1 \leq i \leq N$.

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 $P \in Dom(P)$.

Целью данной работы является описание подхода к оптимизации торговых стратегий, основанного на эволюционных вычислениях, и его распараллеливанию. Представлен генетический алгоритм (ГА), 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 (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.

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 стратегии $S(Gen)$.

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

$$F = D_N = \sum_{m=1}^{N_{br}} (d_m^{br} - Comm), \text{ где } d_m^{br} = \frac{C_m^{sell} - C_m^{buy}}{C_m^{buy}},$$

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 .

Экспериментальные результаты

Предложенный генетический алгоритм с использованием темплейтов был успешно применен для адаптивной оптимизации торговых стратегий, основанных на сле-

дующих, 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 (65535 точек), РАО ЕЭС России (65535 точек), NIKKEI (42620 точек), DJIA - Dow Jones Industrial Average (65535 точек), для периода с 16.04.2006 по 16.04.2007. Мы использовали первые 15000 точек для обучения и остальные точки – для тестирования.

Число итераций и размер популяции выбирались экспериментальным путем, основываясь на параметрах из [7,8]. Значения основных параметров в экспериментах следующие: размер популяции (для однопроцессорной системы) – 16000, число итераций – 200, частота мутации – 0.15, частота кроссовера – 0.7, коммиссионные – 0.001.

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, стратегии также задавались на этом языке. Параллельная реализация генетического алгоритма оптимизации стратегий биржевой торговли выполнена на основе распараллеливания по данным [10] с использованием библиотеки MPI в кластерной системе ИВМиМГ СОРАН НКС-160 (168 процессоров Itanium 2, 1.6 ГГц). Распараллеливание по данным в решаемой задаче сводится к равномерному распределению популяции по процессорам системы. В конце итераций среди всех процессоров выбирается лучшее решение, эта схема минимизирует взаимодействия и позволяет получить почти линейное ускорение. В Табл.1 приведены результаты для оптимизации стратегии с ЕМА для акций ГАЗПРОМа при числе процессоров N , размере популяции на каждом процессоре Pop , времени исполнения $T(сек.)$, полученном ускорении Sp и доходности D_N .

Таблица 1.

N	Pop	$T(сек.)$	Sp	D_N
1	16000	17774	1	6.712
2	8000	8888	1.9997	6.048
4	4000	4444	3.996	5.932
8	2000	2222	7.991	6.195
16	1000	1111	15.998	6.905
32	500	556	31.968	6.879
64	250	278	63.935	6.841
128	125	139	127.87	7.035
160	100	112	158.696	6.919

Используемый генетический алгоритм позволил найти значения параметров торговых стратегий, обеспечивающие увеличение функции суммарной доходности (на 11% - 316% для различных индикаторов, см. Табл.2) по сравнению с известными ранее [1,3,6].

Другая 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

Таблица 2. Увеличение суммарной доходности торговых стратегий

Тип акции \ Индикатор	ЕМА	MACD	RSI	ROC
ГАЗПРОМ	72%	43%	144%	15%
РАО ЕС России	49%	3.8%	44%	11%
NIKKEI	104%	45%	316%	33%
DJIA	41%	35,6%	31%	16%

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 с ЕМА для акций ГАЗПРОМа на системе из двух указанных видеокарт составляет 81 сек., а эквивалентное задание на кластерной системе НКС-160 для 160 процессоров выполняется 82 сек., т.е. для данной задачи эти две системы показывают приблизительно равную производительность.

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 кластерной системе НКС-160 позволяет получить почти линейное ускорение и увеличить значения функции суммарной доходности. Показана целесообразность использования для решения данной задачи и системы на графических процессорах G92 на видеокартах NVIDIA. 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.

Работа выполнена при поддержке РФФИ, проект № 08-01-00857

Литература

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, Charles , 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, V. 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. Monakhov, O. , Monakhova, E. Evolving Templates for Synthesis of Scientific Algorithms. Computational Technologies. N6, 2005, p.3-12.
10. Монахов О.Г., Монахова Э.А. Параллельные системы с распределенной памятью: структуры и организация взаимодействий, Новосибирск: Изд-во СО РАН, 2000.
11. NVIDIA CUDA Programming Guide. Available at http://www.nvidia.com/object/cuda_get.html