

УДК 519.67

DOI 10.17513/snt.39820

ИДЕНТИФИКАЦИЯ ТРЕНДОВ БИРЖЕВЫХ КОТИРОВОК НА ОСНОВЕ УСТОЙЧИВОЙ АДРЕСНОЙ СОРТИРОВКИ

Ромм Я.Е., Турилин А.С.

*Таганрогский институт имени А.П. Чехова (филиал)
ФГБОУ ВО «Ростовский государственный экономический университет»,
Таганрог, e-mail: romm@list.ru, turilinalexandr@yandex.ru*

Для компьютерной идентификации трендов применяется сортировка последовательности оцифрованных данных биржевых валютных торгов. Используется устойчивая адресная сортировка слиянием. С помощью этой сортировки с единственностью определяются одновременно все локально экстремальные элементы (ниже экстремумы) входной последовательности с произвольно фиксированным радиусом локализации. Экстремумы идентифицируются по значению и по индексу местоположения во входном массиве. Выполняется их фильтрация для выделения существенно информативных максимумов и минимумов последовательности. Фильтрация осуществляется путем итеративного повторения идентификации экстремумов с возрастающим на каждой итерации радиусом локализации. Условием остановки итераций является достижение априори заданного числа повторений количества идентифицированных экстремумов. Фильтрация возобновляется с различным (не возрастающим) числом повторений их количества. Окончательно идентифицированные существенно информативные экстремумы объединяются в новый массив в порядке их расположения во входной последовательности оцифрованных данных. В новом массиве идентифицируются максимумы и минимумы с радиусом локализации равным двум отсчетам. Именно эти экстремумы определяют тренд и его разворот. Полученный максимум в конце входной последовательности оцифрованных данных указывает разворот тренда с подъема на спуск, полученный минимум в конце входной последовательности указывает разворот тренда со спуска на подъем. Метод иллюстрируется на исторических данных.

Ключевые слова: биржевые валютные торги, адресная сортировка слиянием, идентификация информативных экстремумов, определение трендов и их разворотов

TREND DETERMINATION OF THE STOCK EXCHANGE QUOTES BASED ON STABLE ADDRESS SORTING

Romm Ya.E., Turilin A.S.

*Taganrog Branch of the Rostov State University of Economics, Taganrog,
e-mail: romm@list.ru, turilinalexandr@yandex.ru*

Sorting of the sequence of the stock exchange trading digitized data is used for the computer trend determination. Stable address merge sorting is used. Using this sort with uniqueness, all locally extreme elements (below mentioned as the extremes) of the input sequence with an arbitrarily fixed localization radius are determined simultaneously. Extremes are identified by the value and by the location index in the input array. Their filtering is performed to highlight significantly informative maxima and minima of the sequence. Filtering is carried out by iteratively repeating the identification of extremes with an increasing localization radius at each iteration. The condition for stopping the iterations is to achieve a priori a given number of repetitions of the number of identified extremes. Filtering resumes with a different (non-increasing) number of repetitions of their number. Finally identified essentially informative extremes are combined into a new array in the order of their location in the input sequence of digitized data. The new array identifies maxima and minima with a localization radius equal to two counts. It is these extremes that determine the trend and its reversal. The received maximum at the end of the input sequence of digitized data indicates a trend reversal from ascent to descent, the received minimum at the end of the input sequence indicates a trend reversal from descent to ascent. The method is based on historical data.

Keywords: stock exchange trading, address merge sorting, identification of informative extremes, determination of trends and their reversals

Рассматривается компьютерная обработка последовательности оцифрованных данных биржевых валютных торгов. Для определенности выбраны данные Fogex на бирже finam [1]. Котировки скачиваются и сохраняются в файл формата .csv. Файл преобразуется во входную числовую последовательность и отображается в графическом формате. Выполняется сортировка этой последовательности. Используется устойчивая адресная сортировка слиянием по матрицам сравнений [2] (ниже сорти-

ровка). Такая сортировка имеет временную сложность $O(N \log_2 N)$, в параллельном варианте – $O(\log_2 N)$, устанавливает взаимно однозначное соответствие входных и выходных индексов сортируемых элементов, что позволяет с единственностью идентифицировать одновременно все локально экстремальные элементы любой числовой последовательности в произвольно фиксированном радиусе локализации [3, 4]. Экстремум идентифицируется по своему значению и местоположению (индексу)

во входной последовательности (в одномерном числовом массиве), что приводит к следующей постановке вопроса. Можно ли с помощью сортировки выделить те экстремальные элементы последовательности оцифрованных данных, которые позволяют распознать и идентифицировать динамическое положение целевого объекта, представленного последовательностью? Применительно к данным биржевых (валютных) торгов вопрос сводится к распознаванию тренда и идентификации его разворота. Ниже рассмотрена возможность прогнозирования разворота тренда и его продолжения на краткосрочный период. Аналогично можно ставить вопрос о прогнозировании на среднесрочный и долгосрочный период. Согласно [5] краткосрочный тренд – от нескольких часов до месяца, среднесрочный – от месяца до года, долгосрочный – более года (средняя продолжительность долгосрочного тренда 2–2,5 года).

Цель работы – раскрыть возможности применения сортировки для распознавания и идентификации целевых объектов по ин-

формационно существенным локально экстремальным элементам последовательности представляющих их оцифрованных данных. Исследование выполняется на примере последовательности данных валютных торгов. В основу подхода полагается иерархическое выделение из входной последовательности локальных экстремумов, информационно существенно связанных с искомым объектом (в данном случае – с трендом и его разворотом).

Описание метода. Свойства сортировки, включая взаимно однозначное соответствие индексов входных и выходных элементов, детально описаны в [3, 4]. Для представления требуемых свойств и программных механизмов с целью применения непосредственно ниже с незначительной модификацией воспроизводится пример из [3]. Для сокращения текста сортировка слиянием эквивалентно заменена сортировкой подсчетом.

Пример 1. Следующая программа (Delphi) идентифицирует все локально минимальные и локально максимальные элементы массива из раздела описания констант.

```

program LokMinMaxIII;
{$APPTYPE CONSOLE}

uses
  SysUtils;

label 2, 22, 222;
const nn=12;
type vect=array [1..nn] of extended; vect0=array [1..nn] of integer;
const b: vect =
(1.20888884+0.0000000000000001, 1.20888884-0.0000000000000001, 1.20888884, -6.304, 1.404,
-1.904, 9.504, 1.504, 14.604, 1.704, -11.804, -11.804-0.0000000000000001);
var i,k,l,n: integer; a,c: vect; e: vect0;
procedure sort (var n: integer; var a,c: vect; var e: vect0);
var i,j,k: integer;
begin
  for i:=1 to n do
  begin
    k:= 0; for j:=1 to i do
    if a[j]<=a[i] then k:=k+1;
    for j:=i+1 to n do
    if a[j]<a[i] then k:=k+1;
    c[k]:=a[i]; e[k]:=i;
  end; end;
begin
  n:=nn; for i:=1 to n do
  a[i]:=b[i]; writeln; writeln; writeln ( ' :6, 'massiv a');
  for i:=1 to n do
  write ( ' :6, a[i]); writeln; writeln; writeln;
  sort (n,a,c,e);writeln; writeln; writeln ( ' :6, 'massiv c');
  for i:=1 to n do
  write ( ' :6, c[i]); writeln; writeln; writeln;
  writeln ( ' :6, 'min eps0=1'); writeln; writeln;
  k:=1; while k<= n do

```

```

begin
for L := 1 to k-1 do if abs(e[k]-e[k- L]) <= 1 then goto 2;
writeln ( '      ',c[k], '      ',e[k]);
2: k:=k+1;
end; writeln; writeln; writeln ( ' :6, 'max eps0=1'); writeln;
k:=1; while k<= n do
begin
for L := 1 to n-k do if abs(e[k]-e[k+ L]) <= 1 then goto 22;
writeln ( '      ',c[k], '      ',e[k]);
22: k:=k+1;
end; writeln; writeln; writeln ( ' :6, 'min eps0=3'); writeln;
k:=1; while k<= n do
begin
for L := 1 to k-1 do if abs(e[k]-e[k- L]) <= 3 then goto 222;
writeln ( '      ',c[k], '      ',e[k]);
222: k:=k+1;
end; writeln; writeln; writeln ( ' :6, 'podstanovka indeksov'); writeln;
for i:=1 to n do write ( ' :2, ' ', i); writeln; writeln ;
for i:=1 to n do write ( ' :2, ' ', e[i]);
readln;
end.

```

Результат работы программы:

massiv a

1.20888884000001E+0000	1.20888883999999E+0000	1.20888884000000E+0000
-6.30400000000000E+0000	1.40400000000000E+0000	-1.90400000000000E+0000
9.50400000000000E+0000	1.50400000000000E+0000	1.46040000000000E+0001
1.70400000000000E+0000	-1.18040000000000E+0001	-1.18040000000000E+0001

massiv c

-1.18040000000000E+0001	-1.18040000000000E+0001	-6.30400000000000E+0000
-1.90400000000000E+0000	1.20888883999999E+0000	1.20888884000000E+0000
1.20888884000001E+0000	1.40400000000000E+0000	1.50400000000000E+0000
1.70400000000000E+0000	9.50400000000000E+0000	1.46040000000000E+0001

min eps0=1

-1.18040000000000E+0001	12
-6.30400000000000E+0000	4
-1.90400000000000E+0000	6
1.20888883999999E+0000	2
1.50400000000000E+0000	8

max eps0=1

1.20888884000000E+0000	3
1.20888884000001E+0000	1
1.40400000000000E+0000	5
9.50400000000000E+0000	7
1.46040000000000E+0001	9

min eps0=3

-1.18040000000000E+0001	12
-6.30400000000000E+0000	4

podstanovka indeksov

1	2	3	4	5	6	7	8	9	10	11	12
12	11	4	6	2	3	1	5	8	10	7	9

Операторы

```

k:=1; while k<= n do
begin
for L := 1 to k-1 do if abs(e[k]-e[k- L]) <= eps0 then goto 2; writeln ( '   ',c[k], '   ',e[k]);
2: k:=k+1; end;
k:=1; while k<= n do
begin
for L := 1 to n-k do if abs(e[k]-e[k+ L]) <=eps00 then goto 22; writeln ( '   ',c[k], '   ',e[k]);
22: k:=k+1; end;

```

идентифицируют все локально минимальные элементы (в дальнейшем локальные минимумы) с радиусом локализации eps0 (радиус измеряется числом отсчетов) в цикле по элементам с входным индексом e[k], а также все локально максимальные элементы (в дальнейшем локальные максимумы) в цикле по элементам с входным индексом e[k] с радиусом локализации eps00. Эти операторы (ниже операторы идентификации экстремумов) работают после однократного выполнения сортировки с любым количеством повторений для любого произвольно заданного радиуса локализации. Взаимно однозначное соответствие входных и выходных индексов устанавливается в процессе сортировки операторами c[k]:=a[i]; e[k]:=i. Экстремум идентифицируется по определению. Смысл оператора идентификации в том, что во входном массиве в окрестности радиуса eps0 индекса e[k] отсортированного элемента c[k]=a[e[k]] не должен находиться индекс ни одного меньшего элемента, отсортированного по неубыванию массива, то есть элемента с индексом e[k-L], иначе говоря, элемента c[k-L]=a[e[k-L]]. Аналогично во входном массиве в окрестности радиуса eps00 индекса e[k] отсортированного элемента c[k]=a[e[k]] не должен находиться индекс ни одного большего элемента отсортированного по неубыванию массива, то есть элемента с индексом e[k+L], иначе говоря, элемента c[k+L]=a[e[k+L]]. Отношение «больше» понимается в смысле упорядочения по неубыванию: элемент тем больше, чем больше его номер в отсортированном массиве. В частности, это так для равных элементов, сортировка устойчива и сохраняет их порядок. Аналогично элемент тем меньше, чем меньше его номер в отсортированном массиве. Сортировка сравнивает, но не преобразует элементы, поэтому не вносит погрешность в предложенную идентификацию экстремумов. Очевидно, что не вносит погрешность и операция вычитания индексов. В результате экстремумы идентифицируются алгоритмически и арифметически безошибоч-

но, причем одновременно все для любого фиксированного радиуса локализации. Так, в примере 1 на входе даны элементы 1.20888884+0.000000000000001, 1.20888884-0.000000000000001, 1.20888884, первый и третий из которых идентифицируются как максимумы, второй – как минимум. При этом экстремумы идентифицируются, как отмечалось, по своему значению и своему местоположению (индексу) во входном массиве, в распечатке индекс указан в строке справа от значения. Существенно используемая в дальнейшем особенность идентификации экстремумов заключается в том, что наибольший элемент в окрестности радиуса eps00 идентифицируется как максимум, даже если он расположен в начале или в конце входного массива. Аналогично в конце или в начале массива идентифицируется минимум в окрестности радиуса eps0. В примере 1 максимален первый элемент 1.20888884+0.000000000000001, минимален последний -11.804-0.000000000000001.

В общем случае изложенным способом идентифицируются все экстремумы числового массива

$$a = (a_1, a_2, \dots, a_n) \tag{1}$$

по массиву индексов, полученному после выполнения сортировки массива (1)

$$e = (e_1, e_2, \dots, e_n), \tag{2}$$

Условие идентификации максимумов с радиусом локализации ε₀ можно представить в виде соотношения [3, 4]

$$\neg \exists \ell \in \overline{1, n-k} : |e_k - e_{k+\ell}| \leq \varepsilon_0,$$

означающего, что неравенство $|e_k - e_{k+\ell}| \leq \varepsilon_0$ не должно выполняться ни при одном ℓ из отрезка $1 \leq \ell \leq n-k$. Условие идентификации минимумов имеет вид

$$\neg \exists \ell \in \overline{1, k-1} : |e_k - e_{k-\ell}| \leq \varepsilon_0,$$

неравенство $|e_k - e_{k-\ell}| \leq \varepsilon_0$ не должно выполняться ни при одном ℓ из отрезка $1 \leq \ell \leq k-1$.

Пусть теперь все идентифицированные с некоторым радиусом локализации $r_{\max} = \text{eps}0$ индексы локальных максимумов массива (1) в порядке неубывания во входном массиве обозначены

$$e_{\max} = (e_{\max 1}, e_{\max 2}, \dots, e_{\max p}), \quad (3)$$

а все идентифицированные с радиусом локализации $r_{\min} = \text{eps}0$ индексы локальных минимумов этой последовательности в порядке неубывания во входном массиве обозначены

$$e_{\min} = (e_{\min 1}, e_{\min 2}, \dots, e_{\min q}). \quad (4)$$

С помощью (3), (4) (элементы-индексы используются как адреса элементов-значений во входном массиве (1)) формируется новый входной массив для сортировки и идентификации именно на основе индексов экстремумов с некоторым новым радиусом локализации. Процесс формирования нового входного массива из локализованных экстремумов включает, во-первых, запись индексов локальных максимумов (3) и индексов локальных минимумов (4) в один новый массив индексов вида

$$e_{\min \max} = (e_{\min 1}, e_{\min 2}, \dots, e_{\min q}, e_{\max 1}, e_{\max 2}, \dots, e_{\max p}); \quad (5)$$

во-вторых, восстановление исходного порядка следования индексов (по возрастанию) из массива (5) в виде нового массива

$$e_{\min \max} = (e_{\min \max 1}, e_{\min \max 2}, \dots, e_{\min \max q+p}); \quad (6)$$

в-третьих, на основе индексов (6) формируется массив значений экстремальных элементов с данным расположением индексов в массиве (1) (элементы (6) используются как адреса элементов-значений в (1)):

$$a_{\min \max} = (a_{\min \max 1}, a_{\min \max 2}, \dots, a_{\min \max q+p}). \quad (7)$$

После преобразований (5)–(7) массив экстремумов (7), расположенных в исходном порядке, принимается за новый входной массив для сортировки и последующей идентификации в нем локальных максимумов (минимумов). Конечная цель таких преобразований – фильтрация экстремумов для выделения существенно информативных максимумов и минимумов последовательности (1). Эту и последующую фильтрацию можно выполнять итерационно до заданного числа повторений количества экстремумов на группе последовательных итераций. Чтобы выделенные экстремумы были существенно информативны в смысле искомой характеристики, в данном случае – в смысле идентификации и разворота тренда, надо применить правильный способ выбора радиуса локализации. Таким обра-

зом, имеется в виду следующая конкретная цель рассматриваемых преобразований: она состоит в нахождении радиуса локализации всех существенно информативных экстремумов. Такой способ заимствуется из [6], он использовался также в [7] и заключается в итерационном подходе к выделению экстремумов с автоматизированным программным выбором радиуса локализации. В этом способе за условие остановки последовательных итераций принимается совпадение количества минимумов (максимумов) на каждой из априори заданного количества последовательных итераций. Итерации начинаются с малого значения радиуса локализации, например с $r_{\min} = 30$. Идентифицируется и запоминается текущее количество минимумов. Затем в исходном массиве (1) с помощью индексов (2) снова идентифицируются минимумы, но уже с увеличенным радиусом локализации, например с $r_{\min} = 31$, и снова запоминается текущее количество минимумов. При этом радиусы локализации отсчитываются именно относительно входных индексов (2) во входном массиве (1). Процесс итеративно продолжается с увеличением радиуса локализации минимумов на каждой итерации до тех пор, пока не произойдет совпадение количества минимумов, запомненных подряд на нескольких последовательных итерациях. Для рассматриваемого процесса идентификации трендов и их разворотов используются эвристически подобранные для данной предметной области числа последовательных совпадений количества идентифицированных локальных минимумов массива (1). На рассматриваемой стадии преобразований выбрано число итераций с совпадением количества минимумов равное 3. Аналогичный оператор автоматизированного программного подбора радиуса локализации применяется к оператору идентификации локальных максимумов. После достижения априори заданного числа совпадений (в данном случае 3) количества экстремумов итерации останавливаются. На выходе данного применения операторов локализации экстремумов в совокупности с оператором автоматического подбора радиуса локализации получают значения экстремумов и места (индексы) их расположения во входной последовательности, которые предварительно принимаются за существенно информативные.

Для последующей идентификации трендов и точек разворота тенденций предварительно описанные итерации повторяются с приводимыми ниже конкретными преобразованиями массивов. Именно на первой итерации на вход поступает одномерный

массив цены закрытия валютной пары вида (1) с одноминутным таймфреймом. Выполняется его сортировка для получения массива индексов вида (2). Затем выполняется идентификация экстремумов во входном массиве с оператором автоматического программного подбора радиуса локализации. В качестве параметра остановки итераций подбора радиуса выступает трехкратное совпадение количества локальных максимумов (минимумов) на каждой из подряд идущих итераций. В результате получаются массивы индексов вида (3) и (4). В соответствии с последующими преобразованиями полученные результаты обозначаются как

$$e^{(1)}_{\max} = (e^{(1)}_{\max 1}, e^{(1)}_{\max 2}, \dots, e^{(1)}_{\max p^{(1)}}) \quad (8)$$

и

$$e^{(1)}_{\min} = (e^{(1)}_{\min 1}, e^{(1)}_{\min 2}, \dots, e^{(1)}_{\min q^{(1)}}) \quad (9)$$

соответственно. Новый входной массив вида (7) обозначается как

$$a^{(1)}_{\min \max} = (a^{(1)}_{\min \max 1}, a^{(1)}_{\min \max 2}, \dots, a^{(1)}_{\min \max q^{(1)}+p^{(1)}}). \quad (10)$$

Далее, относительно (8)–(10) процесс повторяет описанное предварительное выделение существенных экстремумов с той разницей, что имеются три принципиальных отличия. Именно, во-первых, радиус локализации теперь отсчитывается не по индексам входного массива (1), а по индексам сформированного массива (10) (элементы массивов (8), (9) перемещаются в процессе сортировки и используются как адреса идентифицированных элементов входного массива (1)). Во-вторых, начальный радиус в этих итерациях задается равным единице: $r_{\min} = 1, r_{\max} = 1$. В-третьих, в качестве параметра остановки итераций принимается двукратное совпадение количества локальных максимумов (минимумов) на подряд идущих итерациях. На выходе этого процесса получают сформированные массивы индексов локальных максимумов

$$e^{(2)}_{\max} = (e^{(2)}_{\max 1}, e^{(2)}_{\max 2}, \dots, e^{(2)}_{\max p^{(2)}}) \quad (11)$$

и индексов локальных минимумов

$$e^{(2)}_{\min} = (e^{(2)}_{\min 1}, e^{(2)}_{\min 2}, \dots, e^{(2)}_{\min q^{(2)}}), \quad (12)$$

а также формируется новый входной массив

$$a^{(2)}_{\min \max} = (a^{(2)}_{\min \max 1}, a^{(2)}_{\min \max 2}, \dots, a^{(2)}_{\min \max q^{(2)}+p^{(2)}}). \quad (13)$$

К массивам (11)–(13) повторно применяется в точности та же, только что описанная, процедура фильтрации с оператором автоматического подбора радиуса локализации (параметр остановки итераций ра-

вен 2) локальных максимумов. На выходе получается массив индексов локальных максимумов

$$e^{(3)}_{\max} = (e^{(3)}_{\max 1}, e^{(3)}_{\max 2}, \dots, e^{(3)}_{\max p^{(3)}}) \quad (14)$$

и индексов локальных минимумов

$$e^{(3)}_{\min} = (e^{(3)}_{\min 1}, e^{(3)}_{\min 2}, \dots, e^{(3)}_{\min q^{(3)}}), \quad (15)$$

а также формируются два новых входных массива – массив максимумов

$$a^{(3)}_{\max} = (a^{(3)}_{\max 1}, a^{(3)}_{\max 2}, \dots, a^{(3)}_{\max p^{(3)}}) \quad (16)$$

и массив минимумов

$$a^{(3)}_{\min} = (a^{(3)}_{\min 1}, a^{(3)}_{\min 2}, \dots, a^{(3)}_{\min q^{(3)}}). \quad (17)$$

На этом завершается идентификация существенно информативных экстремумов. На выходе представленных итерационных процессов для дальнейшего анализа и идентификации точек разворота тенденции имеются все требуемые данные: значения и индексы существенно информативных локальных экстремумов входной последовательности (14)–(17). Для идентификации тренда и точек его разворота к полученным массивам (14) и (16) применяется оператор локализации максимумов, но уже без итеративного оператора подбора радиуса локализации. Радиус локализации задается фиксированно, непосредственно равным 2. В результате сформируются массивы, содержащие информацию о направлении и развороте тренда. Именно, массив индексов

$$e^{(t)}_{\max} = (e^{(t)}_{\max 1}, e^{(t)}_{\max 2}, \dots, e^{(t)}_{\max p}) \quad (18)$$

будет содержать точки разворота с подъема тренда на его спуск. Аналогично сформируется массив

$$e^{(t)}_{\min} = (e^{(t)}_{\min 1}, e^{(t)}_{\min 2}, \dots, e^{(t)}_{\min q}), \quad (19)$$

содержащий точки разворота тренда со спуска на его подъем.

В значениях элементов с индексами (18), (19) спуск от максимума означает нисходящий тренд, подъем от минимума – восходящий тренд. Если такие экстремумы идентифицировались вблизи входного индекса последнего элемента массива (1), они указывают разворот тренда. Если такой экстремум – максимум, тренд развернется вниз. Если такой экстремум – минимум, тренд развернется вверх.

Ниже приводится фрагмент программы определения тренда и разворота на исторических данных биржевых котировок валютных торгов Forex на бирже Finam. Програм-

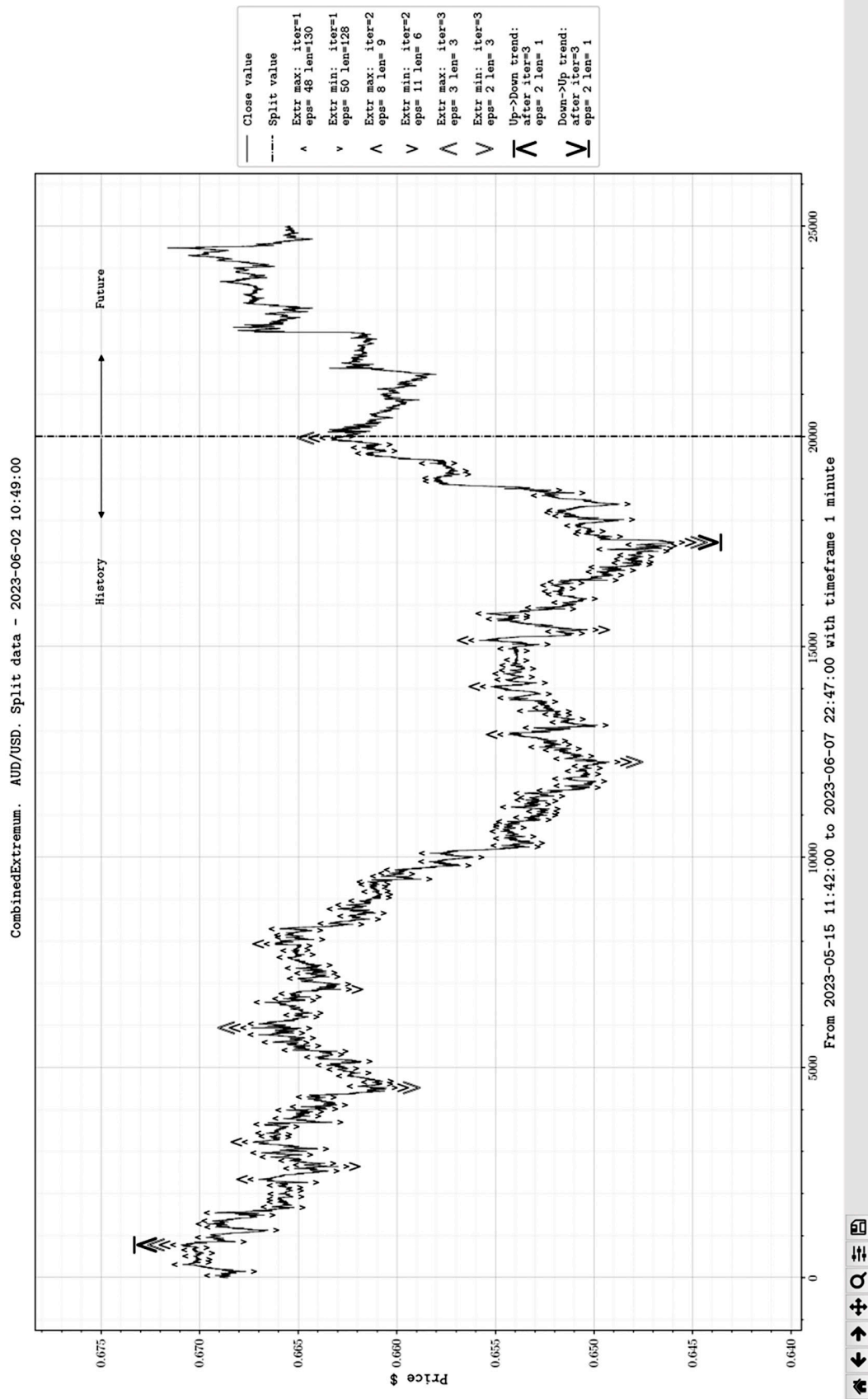
ма написана на языке программирования Python [8], версии 3.10, с использованием дополнительных библиотек: matplotlib [9] – для визуализации результата работы программы; numpy [10] и pandas [11] – для работы с историческими данными котировок, подающимися на вход в файле формата .csv. Во фрагменте программы в роли оператора автоматического подбора радиуса локализации выступает функция coincident(num_iter),

которая в качестве параметра принимает число (num_iter) последовательно идентифицированных количеств локальных экстремумов, функции max_extremum(index, eps) и min_extremum(index, eps) выступают в роли операторов идентификации локальных максимумов и минимумов принимают в качестве параметров: массив индексов (index) и значение радиуса локализации (eps) соответственно:

```
# первый этап
index_1 = merge_arg_sort(close_1)
# остановка подбора радиуса локализации при трехкратном совпадении количества максимумов
max_index_1, max_eps_1 = coincident(3)(max_extremum)(index=index_1, eps=30)
# остановка подбора радиуса локализации при трехкратном совпадении количества минимумов
min_index_1, min_eps_1 = coincident(3)(min_extremum)(index=index_1, eps=30)
# второй этап
index_temp_2 = np.sort(np.hstack((min_index_1, max_index_1)), kind="mergesort")
close_2 = close_1[index_temp_2]
index_2 = merge_arg_sort(close_2)

# остановка подбора радиуса локализации при двукратном совпадении количества максимумов
max_index_2, max_eps_2 = coincident(2)(max_extremum)(index=index_2, eps=1)
# остановка подбора радиуса локализации при двукратном совпадении количества минимумов
min_index_2, min_eps_2 = coincident(2)(min_extremum)(index=index_2, eps=1)
# третий этап
index_temp_3 = np.sort(np.hstack((min_index_2, max_index_2)), kind="mergesort")
close_3 = close_2[index_temp_3]
index_3 = merge_arg_sort(close_3)
# остановка подбора радиуса локализации при двукратном совпадении количества максимумов
max_index_3, max_eps_3 = coincident(2)(max_extremum)(index=index_3, eps=1)
# остановка подбора радиуса локализации при двукратном совпадении количества минимумов
min_index_3, min_eps_3 = coincident(2)(min_extremum)(index=index_3, eps=1)
# Определение точек разворота тренда (с понижения на подъем)
min_eps_trend = 2
min_index_temp_trend = np.sort(min_index_3, kind="mergesort")
min_close_temp_trend = close_3[min_index_temp_trend]
min_index_trend = merge_arg_sort(min_close_temp_trend)
# определение точек разворота тренда с заданным радиусом локализации
min_index_trend_point = min_extremum(index=min_index_trend, eps=min_eps_trend)
min_values_fourth = min_close_temp_trend[min_index_trend_point]
# Определение точек разворота тренда (с подъема на понижение)
max_eps_trend = 2
max_index_temp_trend = np.sort(max_index_3, kind="mergesort")
max_close_temp_trend = close_3[max_index_temp_trend]
max_index_trend = merge_arg_sort(max_close_temp_trend)
# определение точек разворота тренда с заданным радиусом локализации
max_index_trend_point = max_extremum(index=max_index_trend, eps=max_eps_trend)
max_values_trend_point = max_close_temp_trend[max_index_trend_point]
```

Результат работы полной программы графически отображен на рисунке.



Идентификация трендов и разворотов тенденций

Работа программы соответствует описанному алгоритму. Программа применялась к историческим данным биржевых котировок валютной пары AUD/USD за период с 2023-05-15 11:42 по 2023-06-07 22:47 с таймфреймом равным 1 мин. Для моделирования работы в реальном времени, входные данные разделены на исторические (History) и будущие (Future). На рисунке это разделение обозначено штрихпунктирной вертикалью (Split value), слева от нее исторические, справа – будущие данные. Пояснительная информация отражена на легенде справа от графика. Входные данные цены закрытия валютной пары обозначены непрерывной линией (Close value). На первой итерации идентифицировано 130 существенно информативных максимумов (на графике отмечены символом \wedge) с автоматически подобранным радиусом локализации $\text{eps}=48$ и 128 минимумов (на графике отмечены символом \vee) с автоматически подобранным радиусом локализации $\text{eps}=50$. На легенде это отображено соответственно как $\wedge\text{Etr max: iter} = 1 \text{ eps} = 48, \text{ len} = 130$ и $\vee\text{Etr min: iter} = 1 \text{ eps} = 50, \text{ len} = 128$ с соответствующими маркерами. После второй итерации $\wedge\text{Etr max: iter} = 2 \text{ eps} = 8, \text{ len} = 9$ и $\vee\text{Etr min: iter} = 2 \text{ eps} = 11, \text{ len} = 6$ (на графике полученные экстремумы отмечены теми же символами, но большего размера: \wedge, \vee). Третья итерация оставила только существенно информативные значения и места расположения в исходном массиве локальных максимумов (минимумов) – $\wedge\text{Etr max: iter} = 3 \text{ eps} = 3, \text{ len} = 3$ и $\vee\text{Etr min: iter} = 3 \text{ eps} = 2, \text{ len} = 3$ (на графике экстремумы отмечены соответственно увеличенными символами \wedge и \vee).

После проведенной фильтрации на выделенных существенно информативных экстремумах (14), (15) и их значениях (16), (17) программно идентифицированы точки разворота тенденций (18) и (19). Разворот с подъема на понижение (на графике отмечен символом $\bar{\wedge}$) идентифицирован как точка разворота после третьей итерации $\bar{\wedge}$ Up->Downtrend: after iter = 3 eps = 2 len = 1. Разворот тренда с понижения на подъем (на графике отмечен символом $\bar{\vee}$) идентифицирован как точка разворота после третьей итерации $\bar{\vee}$ Down->Uptrend: after iter = 3 eps = 2 len = 1.

Результаты исследования и их обсуждение

Согласно рисунку, на краткосрочный период (Future) спрогнозировано продолжение тренда с понижения на подъем. Предложенный метод правильно прогнозирует тренд

и точки разворота в целом, с достаточно высокой достоверностью. Подсчет вероятности правильного прогноза является предметом самостоятельного исследования, однако на большом количестве численных экспериментов метод почти всегда дает правильный результат. Его применение целесообразно в случае обработки данных большого числа биржевых котировок. На основании экспериментов метод с большой вероятностью дает достоверные результаты при прогнозировании на среднесрочные и долгосрочные периоды, в этом случае он соответственно модифицируется. В частности, необходимо изменить таймфрейм входных данных. Так, для прогноза на среднесрочный период таймфрейм выбирается равным 15–30 мин, на долгосрочный – от часа до суток. Изложенный подход может иметь значение для оценок сугубо экономического состояния предприятия на основе показаний биржевых торгов акциями предприятия. Если дополнить метод соответствующими трактовками конкретной предметной области, то на его основе можно выделять, распознавать и идентифицировать целевые объекты, в частности, по оцифрованной последовательности локационных сигналов. Апробация численных и программных экспериментов в некоторых из данных аспектов представлена в [12, 13], конструктивное применение сортировки для распознавания образов описано в [14, 15].

Заключение

Подход к компьютерной обработке оцифрованных данных биржевых валютных торгов представлен на примере обработки данных Forex на бирже Finam. Устойчивая адресная сортировка одномерного массива данных позволяет с единственностью идентифицировать одновременно все локальные экстремумы с любым радиусом локализации по значению и индексу местоположения. Выделяются наиболее информативные экстремумы, идентифицирующие тренд и его разворот на краткосрочный период. С соответствующим видоизменением подход допускает применение сортировки для распознавания и идентификации целевых объектов по существенно информативным локально экстремальным элементам последовательности оцифрованных данных.

Список литературы

1. Forex: курсы валют онлайн. [Электронный ресурс]. URL: <https://www.fxclub.org/markets/forex> (дата обращения: 02.09.2023).
2. Ромм Я.Е. Параллельная сортировка слиянием по матрицам сравнений. I // Кибернетика и системный анализ. 1994. № 5. С. 3–23.

3. Ромм Я.Е. Идентификация нулей и экстремумов функций на основе сортировки с приложением к анализу устойчивости. I. Случай одной действительной переменной // Современные наукоемкие технологии. 2020. № 6-1. С. 79–97. DOI: 10.17513/snt.38075.
4. Ромм Я.Е. О границах идентификации корней полиномов на основе устойчивой адресной сортировки // Современные наукоемкие технологии. № 12-1. 2021. С. 84–108. DOI: 10.17513/snt.38959.
5. Словарь инвестора. Жизненный цикл тренда. [Электронный ресурс]. URL: <https://beststocks.ru/dictionary/life-trend> (дата обращения: 08.10.2023).
6. Рюмин О.Г. Распараллеливаемая идентификация растеризованного изображения на основе экстремальных признаков замкнутого контура // Известия ЮФУ. Технические науки. 2008. № 7 (84). С. 59–66.
7. Тренкеншу А.И. Программное обеспечение для компьютерной идентификации ключевых фигур и предсказания тенденций графиков биржевых котировок по экстремальным признакам на основе алгоритмов сортировки. Свидетельство о государственной регистрации программы для ЭВМ № 2014610934 от 20.01.2014.
8. Python: is a programming language that lets you work quickly and integrate systems more effectively. [Электронный ресурс]. URL: <https://www.python.org> (дата обращения: 08.10.2023).
9. Matplotlib: visualization with Python [Электронный ресурс]. URL: <https://matplotlib.org> (дата обращения: 08.10.2023).
10. NumPy: the fundamental package for scientific computing with Python. [Электронный ресурс]. URL: <https://numpy.org> (дата обращения: 08.10.2023).
11. Pandas: is a fast, powerful, flexible, and easy to use opensource data analysis and manipulation tool, built on top of the Python programming language. [Электронный ресурс]. URL: <https://pandas.pydata.org> (дата обращения: 08.10.2023).
12. Ромм Я.Е., Дордопуло А.И., Заярный В.В. Программная локализация нулей многочленов с приложением к идентификации объектов по данным гидроакустической локации. Таганрог: Изд-во ТГПИ, 2005. 217 с.
13. Ромм Я.Е., Корякин А.Б. Построение признаков распознавания с применением сортировки для обработки гидроакустических сигналов // Известия ЮФУ. Технические науки. 2010. № 5. С. 95–102.
14. Ромм Л.Я. Методы распознавания плоских контурных и внутриконтурных изображений на основе сортировки экстремальных признаков и подстановки индексов // Известия ЮФУ. Технические науки. 2009. № 5. С. 23–30.
15. Somshubra Majumdar, Ishaan Jain, Kunal Kukreja, Kiran Bhowmick. Adaptive Sorting Using Machine Learning // (IJCISIT) International Journal of Computer Science and Information Technologies. 2016. Vol. 7 (2). P. 490–493.