

○ 「 任意指標の最適化手法 」

・アメンボです、

今回は、「任意の指標」を最適化する手法を解説します、
MT5 には専用の機能が存在するのですが、MT4 では専用機能はありません。
本稿で MT4 での実現手法を解説しておきながら、アメンボはまだ有効な使用例を
思いついていないので、諸兄においては色々試して頂きたく公開します。

・手法自体はそれほど難しいものではありませんが、エクセル・マクロの基本が理解できる
必要があります。

MQL4 の中だけでも、データ並べ替え（ソート）は可能なのですが、拡張性やグラフ化を考えて、
エクセルでデータ処理をすることにしました。（エクセル・マクロは覚える価値があります）

<同時掲載資料>・・・ダウンロード用

- ・MQL4 コード；
 - ①動作確認用の基本コード 「 **Skeleton_opt.mq4** 」
 - ②トライした E A 「 **Bollin_EA_08_opt.mq4** 」
 - ③出力ファイル（参考添付） 「 **opt_data.txt** 」
- ・エクセル VBA マクロ（エクセル 2003）； 「 **Sort_vba.xls** 」

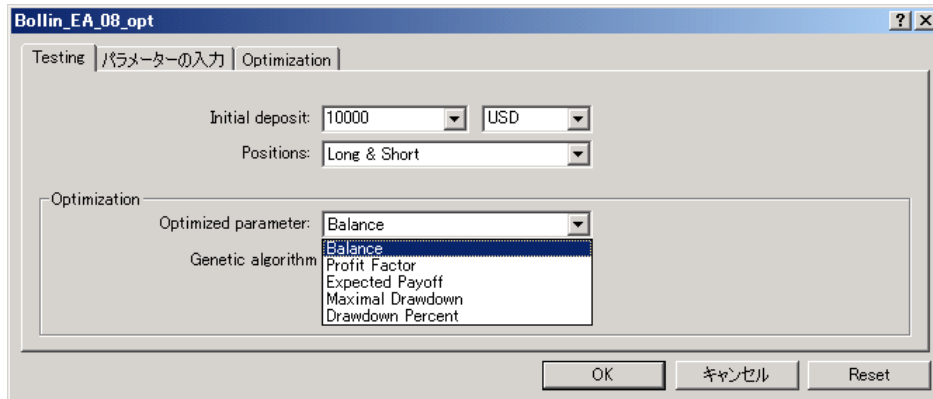
上記の「②」が出力するファイル「opt_data.txt」のデータを読取って処理します、
「tester/files」フォルダから「opt_data.txt」を、「Sort_vba.xls」の在るフォルダーに
コピーしてからマクロを動かしてください。

目次：	1. 「任意指標の最適化」の意味	・・・	2 頁
	(1) MT4 で使用可能な指標		
	(2) 本稿でトライする任意指標		
	2. 骨格となる基本 MQL4 コード	・・・	3 頁
	(1) 最適化 (Optimization) 処理の再確認		
	(2) 動作確認用の基本コード		
	(3) 最適化条件		
	(4) int での組合せを実施した場合（ファイル出力内容）		
	(5) double での組合せを実施した場合（ファイル出力内容）		
	3. 「Bollin_EA_08.mq4」への適用例	・・・	8 頁
	(1) トライする内容		
	(2) コード中の主要部分（Bollin_EA_08.mq4 からの相違点）		
	(3) 最適化実施		
	(4) ファイル出力例（opt_data.txt）		
	(5) コード（Bollin_EA_08_opt.mq4）		
	4. エクセルによるデータ処理例	・・・	20 頁
	(1) 処理フロー概要		
	(2) 実施例		
	(3) マクロ・コード (VBA)		

1. 「任意指標の最適化」の意味

(1) MT4 で使用可能な指標

- ・通常の最適化では、下記「Balance（資産）」の最適化を使う場合が殆どだと思います。つまり、「Balance（資産）」が最大になるような「条件（パラメータの組合せ）」を探索しているはずですが。



- ・しかし、観れば判る通り、MT4 は標準で 5 個（下記）の指標の最適化を図ることが出来ます。「Balance、Profit Factor、Expected Payoff、Maxmal Drawdown、Drawdown Percent」

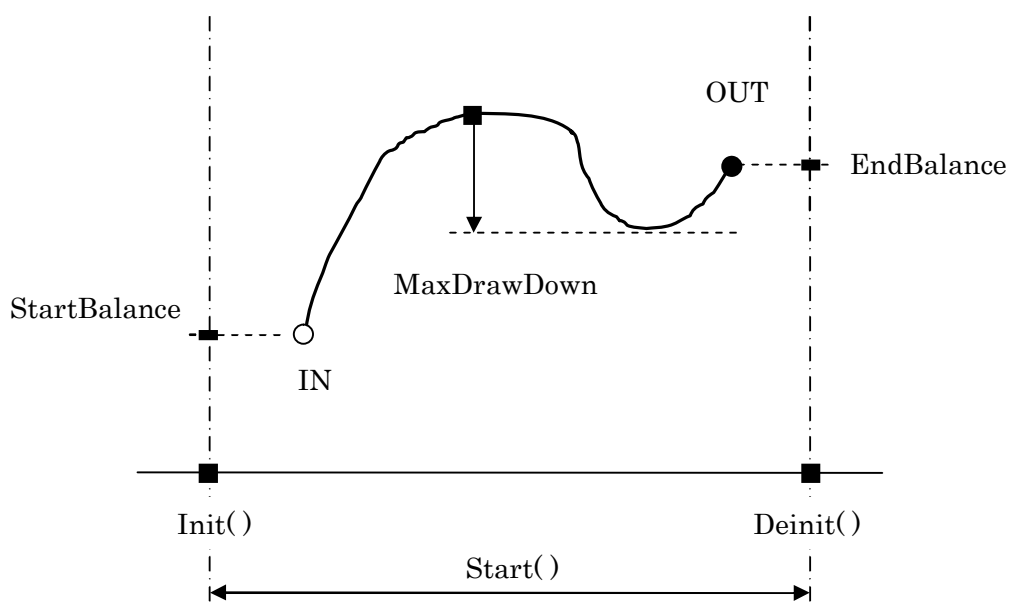
(2) 本稿でトライする任意指標

- ・本当に必要な指標か否かは別として、「任意の指標」を例えば下記とします。

$$Criterion = \frac{(EndBalance - StartBalance)}{MaxDrawDown} = \frac{(AccountBalance() - StartBalance)}{MaxDrawDown}$$

- Criterion ; 最適化を図る指標
- EndBalance(=AccountBalance()) ; テスト後の資産 AccountBalance()
- StartBalance ; テスト前の資産
- MaxDrawDown ; テスト中の最大ドローダウン

概念図；



この指標（判断指標・基準；Criterion）を最適化する手法を解説します。

2. 骨格となる基本 MQL4 コード

(1) 最適化 (Optimization) 処理の再確認

- 最適化を実施すると、「extern 宣言したパラメータ」を少しずつ変えた場合の「指標値」で EA を実行していき、その結果を表示します。

そのデータ中から、「指標値」が最適値（通常は最大か最小）になる「パラメータ」を選んで EA の最適設定ちとすることが「最適化」です。(判りきったこと!?)

- 最適化処理のイメージ;

下記のように、[init()・strat()・deinit()] を一サイクルとして、繰返していきます。

実施番号	EA 処理	extern int parameter 5 から 10 まで「1」ステップ	指標 (例 ; Balance)
1	init()		
	start()	5	
	deinit()		評価
	↓		
2	init()		
	start()	6(=5+1)	
	deinit()		評価
	↓		
3	init()		
	start()	7(=6+1)	
	deinit()		評価
	↓		
	↓		
6	init()		
	start()	10(=9+1)	
	deinit()		評価

(2) 動作確認用の基本コード

まず、最適化のイメージを容易にするためのコード実施例を示します。

- 名称 ; 「 **Skeleton_opt.mq4** 」

結果のデータは「tester/files/ opt_data1.txt」に出力する

(ファイルが存在しない場合は、自動で新規に作成します)

- 出力データは「int」と「double」のどちらかをコメントアウト「/* */」で選択してください

```
//+-----+
//|                               Skeleton_opt.mq4 |
//|                               amenbo |
//|                               |
//+-----+
#property copyright "amenbo"
/*
//各データが double の場合-----
extern double param1=0;//0.1 ずつ増加させるには double であること
```

```

extern double param2=0;//int では「1」 ずつのステップ 制限となる
//
double Criterion=0;
*/
//-----
//各データが int の場合
extern int param1=1;//0.1 ずつ増加させるには double であること
extern int param2=5;//int では「1」 ずつのステップ 制限となる
//
int Criterion=0;
//-----
//
int init()
{
    if(IsTesting() && IsOptimization())
    {
        //
        if((param1==1) && (param2==5))
        {
            //ファイル内容をクリアする
            int h1=FileOpen("opt_data1.txt",FILE_CSV | FILE_WRITE,');
            FileClose(h1);
            //ファイルへ「タイトル項目」を書き出す
            int h2=FileOpen("opt_data1.txt",FILE_CSV | FILE_WRITE,');
            /*
            //各データが double の場合
            FileWrite(h2,"パ ラム 1"+"%t%t", "パ ラム 2"+"%t%t%t", "指標");
            */
            //各データが int の場合
            FileWrite(h2,"パ ラム 1"+"%t", "パ ラム 2"+"%t", "指標");

            //
            FileClose(h2);
            //
        }
        //
    }
    //
    //
    return(0);
}
//
int deinit()
{
    //Counter=Counter+1;
    if(IsTesting() && IsOptimization())
    {
        //動作確認用コード
        //
        //指標の計算例
        Criterion=param1+param2;
        //
        int h3=FileOpen("opt_data1.txt",FILE_CSV | FILE_WRITE | FILE_READ,');
        FileSeek(h3, 0, SEEK_END);
        /*
        //各データが double の場合
        FileWrite(h3,param1+"%t%t",param2+"%t%t",Criterion);
        */
        //各データが int の場合
    }
}

```

```

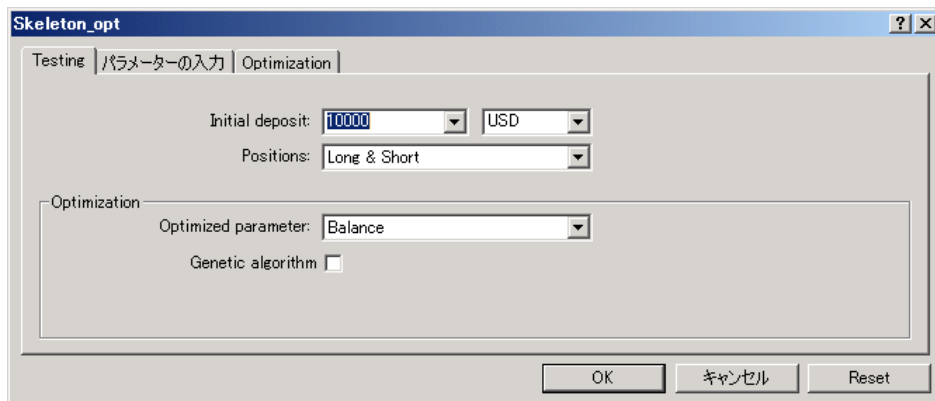
FileWrite(h3,param1+"¥t",param2+"¥t",Criterion);

FileClose(h3);
}
return(0);
}
//
int start()
{
//
if(IsTesting() && IsOptimization())
{
//動作確認用コード（注意）各 Tick ごとに繰り返す処理であること！
//処理記述
}
//
return(0);
}
//

```

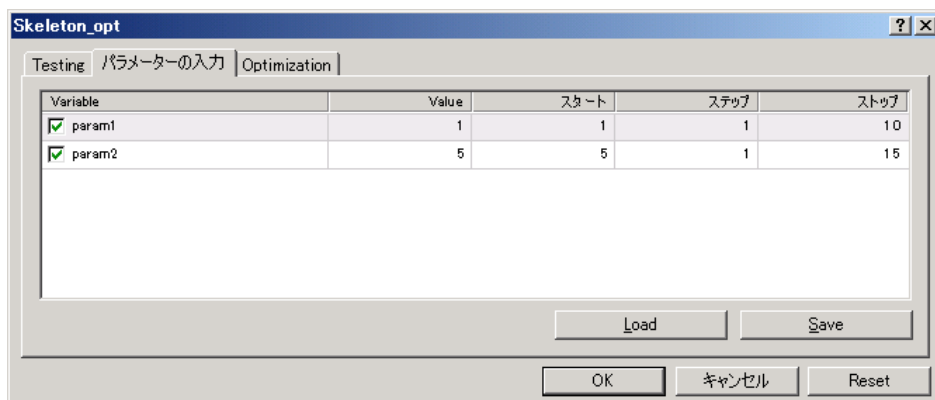
(3) 最適化条件

※int、double で共通



例 ; 「param1」

スタート「1」、ステップ「1」、ストップ「10」に設定します



- ・出力ファイル (opt_data1.txt) には、「param1」と「param2」の最適化後（「総当り」組み合わせ）の結果を出力します。

(4) intでの組合せを実施した場合 (ファイル出力内容)

出力先ファイル ; 「opt_data1.txt」 (tester/files フォルダ中)

パラム1	パラム2	指標
1	,5	,6
2	,5	,7
3	,5	,8
4	,5	,9
5	,5	,10
6	,5	,11
7	,5	,12
8	,5	,13
9	,5	,14
10	,5	,15
1	,6	,7
2	,6	,8
3	,6	,9
4	,6	,10
5	,6	,11
6	,6	,12
7	,6	,13
8	,6	,14
9	,6	,15
10	,6	,16
1	,7	,8
...
5	,14	,19
6	,14	,20
7	,14	,21
8	,14	,22
9	,14	,23
10	,14	,24
1	,15	,16
2	,15	,17
3	,15	,18
4	,15	,19
5	,15	,20
6	,15	,21
7	,15	,22
8	,15	,23
9	,15	,24
10	,15	,25

(5) double での組合せを実施した場合 (ファイル出力内容)

出力先ファイル ; 「opt_data1.txt」 (teister/files フォルダ中)

パラム 1	パラム 2	指標
1.00000000	,5.00000000	,6
2.00000000	,5.00000000	,7
3.00000000	,5.00000000	,8
4.00000000	,5.00000000	,9
5.00000000	,5.00000000	,10
6.00000000	,5.00000000	,11
7.00000000	,5.00000000	,12
8.00000000	,5.00000000	,13
9.00000000	,5.00000000	,14
10.00000000	,5.00000000	,15
1.00000000	,6.00000000	,7
2.00000000	,6.00000000	,8
3.00000000	,6.00000000	,9
4.00000000	,6.00000000	,10
5.00000000	,6.00000000	,11
6.00000000	,6.00000000	,12
7.00000000	,6.00000000	,13
8.00000000	,6.00000000	,14
9.00000000	,6.00000000	,15
10.00000000	,6.00000000	,16
1.00000000	,7.00000000	,8
.....
8.00000000	,13.00000000	,21
9.00000000	,13.00000000	,22
10.00000000	,13.00000000	,23
1.00000000	,14.00000000	,15
2.00000000	,14.00000000	,16
3.00000000	,14.00000000	,17
4.00000000	,14.00000000	,18
5.00000000	,14.00000000	,19
6.00000000	,14.00000000	,20
7.00000000	,14.00000000	,21
8.00000000	,14.00000000	,22
9.00000000	,14.00000000	,23
10.00000000	,14.00000000	,24
1.00000000	,15.00000000	,16
2.00000000	,15.00000000	,17
3.00000000	,15.00000000	,18
4.00000000	,15.00000000	,19
5.00000000	,15.00000000	,20
6.00000000	,15.00000000	,21
7.00000000	,15.00000000	,22
8.00000000	,15.00000000	,23
9.00000000	,15.00000000	,24
10.00000000	,15.00000000	,25

3. 「Bollin_EA_08.mq4」への適用例

(1) トライする内容

①「1. (2)」で解説した下記の指標を使います

$$Criterion = \frac{(EndBalance - StartBalance)}{MaxDrawDown} = \frac{(AccountBalance() - StartBalance)}{MaxDrawDown}$$

Criterion	; 最適化を図る指標
EndBalance(=AccountBalance0)	; テスト後の資産 (AccountBalance0)
StartBalance	; テスト前の資産
MaxDrawDown	; テスト中の最大ドローダウン

②最適化するパラメータは、下記と2つとします

売買 (ポジション) IN する場合の条件;

IKA	・・・「Bollin ; ボリンジャー・バンド」の幅の最大値
IZYOU	・・・「Bollin ; ボリンジャー・バンド」の幅の最小値

※パラメータ数が多いと、時間が掛かって仕方がないので、2つに抑えた。

(2) コード中の主要部分 (Bollin_EA_08.mq4 からの相違点)

・コード名称「Bollin_EA_08_opt.mq4」とします

「Bollin_EA_08.mq4」からの相違部 (青書部) のみを抜き出し、解説します。

①「グローバル宣言」部分

```

=====
// 「任意指標の最適化」実施用
//最適化する指標;
//Criterion=(AccountBalance0-StartBalance)/MaxDrawDown とする
extern double IKA=0.2; //start=0.2、step=0.1 とする
extern double IZYOU=0.0;//start=0.0、step=0.1 とする
//
double StartBalance;
double MaxEquity;
double MaxDrawDown;
double Criterion;
=====
.....
=====
// 「任意指標の最適化」実施用
////extern double IKA=0.4;
////extern double IZYOU=0.2;
=====

```

※必要な「新規変数」の宣言と、「IKA、IZYOU」の設定変更

② 「init0関数」部分

```

int init0
{
    . . . . .
    // 「任意指標の最適化」実施用
    if(IsTesting() && IsOptimization())
    {
        StartBalance=AccountBalance();
        Criterion=0;
        MaxEquity=0;
        MaxDrawDown=0;
        //
        if((IKA==0.1) && (IZYOU==0.0))
        {
            //ファイル内容をクリアする
            int h1=FileOpen("opt_data.txt",FILE_CSV | FILE_WRITE,');
            FileClose(h1);
            //ファイルへ「タイトル項目」を書き出す
            int h2=FileOpen("opt_data.txt",FILE_CSV | FILE_WRITE,');
            FileWrite(h2,"IZYOU"+"¥t¥t¥t","IKA"+"¥t¥t¥t","指標");
            FileClose(h2);
            //
        }
    }
    . . . . .
    return(0);
}

```

※変数・ファイルの初期化と、ファイルへ「タイトル」を出力

③ 「deinit0関数」部分

```

int deinit0
{
    // 「任意指標の最適化」実施用
    if(IsTesting() && IsOptimization())
    {
        if(AccountEquity>MaxEquity)
        {
            MaxEquity=AccountEquity();
        }
        //
        if((MaxEquity-AccountEquity())>MaxDrawDown)
        {
            MaxDrawDown=MaxEquity-AccountEquity();
        }
        //最適化する指標値=Criterion
        Criterion=(AccountBalance()-StartBalance)/MaxDrawDown;
        //ファイルへ書き出し
        int h3=FileOpen("opt_data.txt",FILE_CSV | FILE_WRITE | FILE_READ,');
        FileSeek(h3, 0, SEEK_END);
        FileWrite(h3,IZYOU+"¥t¥t",IKA+"¥t¥t",Criterion);
        FileClose(h3);
    }
    //
    return(0);
}

```

※「一つのパラメータ組合せ」に対する「指標の計算結果；EA 実施結果」をファイルへ出力する。

④ 「start()関数」部分

```

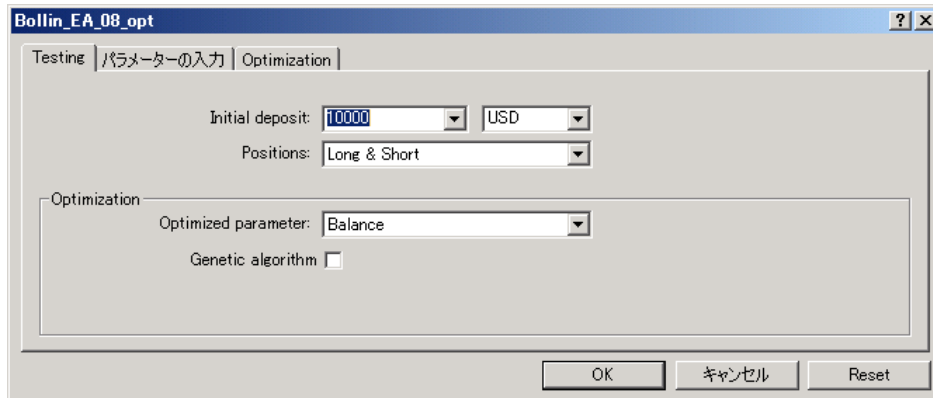
int start()
{
//=====
// 「任意指標の最適化」実施用
if(IsTesting() && IsOptimization())
{
    if(AccountEquity>MaxEquity)
    {
        MaxEquity=AccountEquity();
    }
    //
    if((MaxEquity-AccountEquity())>MaxDrawDown)
    {
        MaxDrawDown=MaxEquity-AccountEquity();
    }
}
//=====
}

```

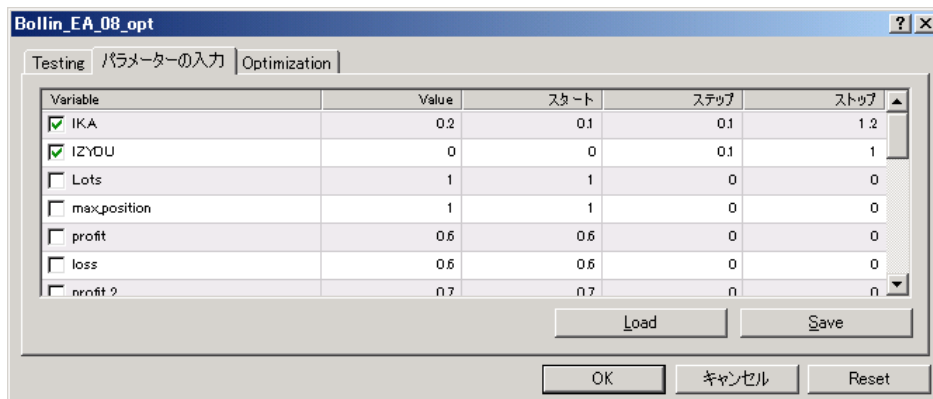
※各「Tick」（売買成立）ごとに、「指標」を再計算して更新する。
 （「一つのパラメータ組合せ」の範囲内で計算する）

(3) 最適化実施

①実施条件

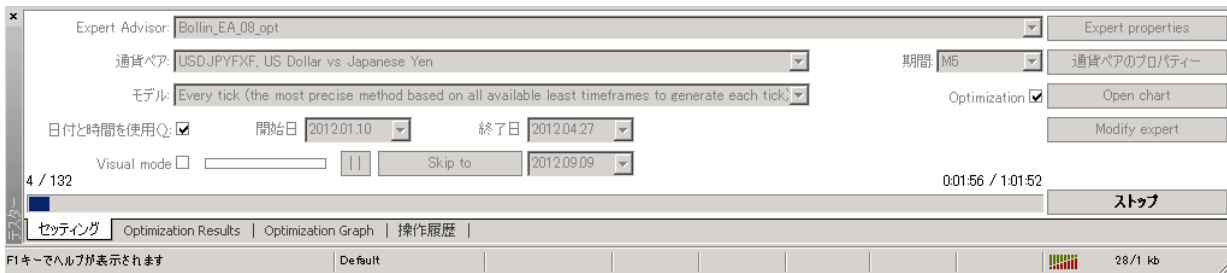


IKA (0~1.2 ; ステップ 0.1) と IZYOU (0~1.0 ; ステップ 0.1) の組合せ

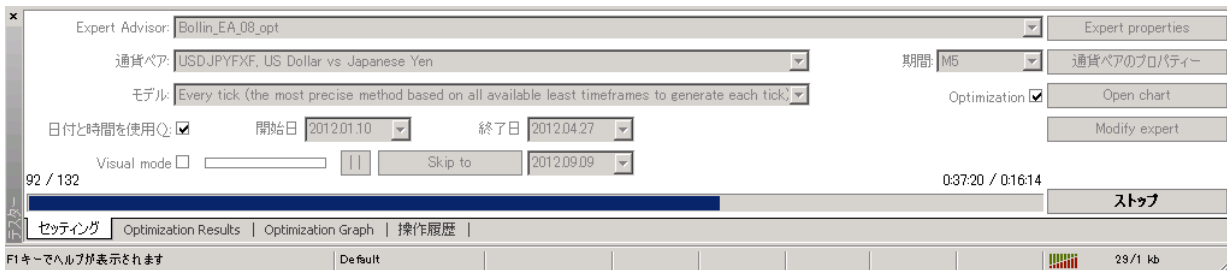


②実施過程

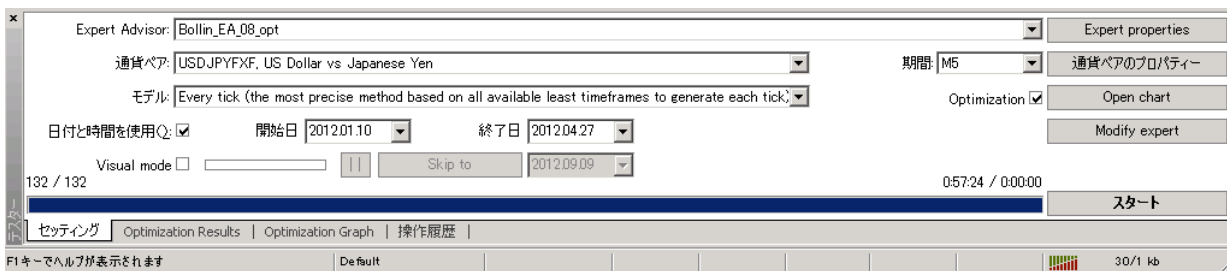
最適化スタート直後



途中



終了



※解説

- ・「4/132」; 総組合せ数「132」中、「4」まで終わった
- ・「0:01:56/1:01:52」; 「1分56秒」経過した、終了まであと「1時間01分52秒」と予測
(終了までの時間はMT4が予測しているが、結構変化する)

(4) ファイル出力例 (opt_data.txt)

IZYOU	,IKA	,指標
0.00000000	,0.20000000	, -0.3759658
0.00000000	,0.30000000	, 2.73092852
0.00000000	,0.40000000	, 6.65320383
0.00000000	,0.50000000	, 3.04744081
0.00000000	,0.60000000	, 3.2649805
0.00000000	,0.70000000	, 2.37816825
0.00000000	,0.80000000	, 2.51484047
0.00000000	,0.90000000	, 1.91984359
0.00000000	,1.00000000	, 1.91984359
0.00000000	,1.10000000	, 1.91984359
0.00000000	,1.20000000	, 1.91984359
0.10000000	,0.20000000	, -0.3759658
0.10000000	,0.30000000	, 2.73092852
0.10000000	,0.40000000	, 6.65320383
0.10000000	,0.50000000	, 3.04744081
0.10000000	,0.60000000	, 3.2649805
0.10000000	,0.70000000	, 2.37816825
0.10000000	,0.80000000	, 2.51484047
0.10000000	,0.90000000	, 1.91984359
0.10000000	,1.00000000	, 1.91984359
0.10000000	,1.10000000	, 1.91984359
0.10000000	,1.20000000	, 1.91984359
0.20000000	,0.30000000	, 3.22170766
0.20000000	,0.40000000	, 7.07505993
0.20000000	,0.50000000	, 3.93754337
0.20000000	,0.60000000	, 4.6889042
0.20000000	,0.70000000	, 3.20537752
0.20000000	,0.80000000	, 3.37678603
0.20000000	,0.90000000	, 2.48590001
0.20000000	,1.00000000	, 2.48590001
0.20000000	,1.10000000	, 2.48590001
0.20000000	,1.20000000	, 2.48590001
0.30000000	,0.40000000	, 5.95398567
0.30000000	,0.50000000	, 3.11520212
0.30000000	,0.60000000	, 2.84500496
0.30000000	,0.70000000	, 1.96950537
0.30000000	,0.80000000	, 2.12314056
0.30000000	,0.90000000	, 2.12314056
0.30000000	,1.00000000	, 2.12314056
0.30000000	,1.10000000	, 2.12314056
0.30000000	,1.20000000	, 2.12314056
0.40000000	,0.50000000	, 0.45745333
0.40000000	,0.60000000	, 0.16779829
0.40000000	,0.70000000	, -0.07158529
0.40000000	,0.80000000	, 0.00692284
0.40000000	,0.90000000	, 0.00692284
0.40000000	,1.00000000	, 0.00692284
0.40000000	,1.10000000	, 0.00692284
0.40000000	,1.20000000	, 0.00692284
0.50000000	,0.60000000	, 0.54277368
0.50000000	,0.70000000	, 0.20969038
0.50000000	,0.80000000	, 0.31791605
0.50000000	,0.90000000	, 0.08608355
0.50000000	,1.00000000	, 0.08608355
0.50000000	,1.10000000	, 0.08608355
0.50000000	,1.20000000	, 0.08608355
0.60000000	,0.70000000	, -0.30382378
0.60000000	,0.80000000	, -0.22926163
0.60000000	,0.90000000	, -0.39669873
0.60000000	,1.00000000	, -0.39669873
0.60000000	,1.10000000	, -0.39669873
0.60000000	,1.20000000	, -0.39669873
0.70000000	,0.80000000	, -0.719612
0.70000000	,0.90000000	, -0.82564572
0.70000000	,1.00000000	, -0.82564572
0.70000000	,1.10000000	, -0.82564572
0.70000000	,1.20000000	, -0.82564572
0.80000000	,0.90000000	, -1
0.80000000	,1.00000000	, -1
0.80000000	,1.10000000	, -1
0.80000000	,1.20000000	, -1
0.90000000	,1.00000000	, -0.56619174
0.90000000	,1.10000000	, -0.56619174
0.90000000	,1.20000000	, -0.56619174
1.00000000	,1.10000000	, -0.85091964
1.00000000	,1.20000000	, -0.85091964

(5) コード (Bollin_EA_08_opt.mq4)

```

//+-----+
//|          Bollin_EA_08_opt.mq4  |
//|          2012.09.09 amenbo  |
//+-----+
#define Magic_ID 1930
//
//
//=====
// 「任意指標の最適化」実施用
//最適化する指標;
//Criterion=(AccountBalance0-StartBalance)/MaxDrawDown とする
extern double IKA=0.2; //start=0.2、step=0.1 とする
extern double IZYOU=0.0;//start=0.0、step=0.1 とする
//
double StartBalance;
double MaxEquity;
double MaxDrawDown;
double Criterion;
//=====
extern double Lots=1;
extern int max_position=1;//最大保有ポジション数
// 損益設定
// 「①②」は最適化の実施後での設定値
extern double profit=0.60;
extern double loss=0.60;
extern double profit_2=0.70;
extern double loss_2=0.50;
// ①Bollin の幅
extern int period_bollin=40;
//=====
// 「任意指標の最適化」実施用
///extern double IKA=0.4;
///extern double IZYOU=0.2;
//=====
// ②フィルター設定
extern int shortPeriod_buy=50;
extern int mediumPeriod_buy=120;
extern double short_buy=-0.0044;
extern double long_buy=-0.0006;
//    ・ ・ sell 専用
extern int shortPeriod_sell=35;
extern int mediumPeriod_sell=180;
extern double short_sell=0.0044;
extern double long_sell=0.0006;
//-----
// ③トレンド判断
extern int trendPeriod=300;//70
extern double _up=-0.00024;//-0.0004
extern double _down=0.00024;
extern double div_=0.04;
//=====
static datetime lastbar;
datetime in_time,out_time;
int barsTotal;
//売買シグナル判定用の配列データ
double EMA_[10];
double Bol_hi[10];
double Bol_lo[10];
double Price_01[10000];
double Trend_[10000];
//////////
int init()
{

```

```

//
lastbar=Time[1];
barsTotal = Bars;
in_time=Time[20];
//
// 「任意指標の最適化」 実施用
if(IsTesting() && IsOptimization())
{
    StartBalance=AccountBalance();
    Criterion=0;
    MaxEquity=0;
    MaxDrawDown=0;
    //
    if((IKA==0.1) && (IZYOU==0.0))
    {
        //ファイル内容をクリアする
        int h1=FileOpen("opt_data.txt",FILE_CSV | FILE_WRITE,');
        FileClose(h1);
        //ファイルへ「タイトル項目」を書き出す
        int h2=FileOpen("opt_data.txt",FILE_CSV | FILE_WRITE,');
        FileWrite(h2,"IZYOU"+"¥t¥t¥t","IKA"+"¥t¥t¥t","指標");
        FileClose(h2);
        //
    }
}
//
//
return(0);
}
//
int deinit()
{
    // 「任意指標の最適化」 実施用
    if(IsTesting() && IsOptimization())
    {
        if(AccountEquity()>MaxEquity)
        {
            MaxEquity=AccountEquity();
        }
        //
        if((MaxEquity-AccountEquity())>MaxDrawDown)
        {
            MaxDrawDown=MaxEquity-AccountEquity();
        }
        //最適化する指標値=Criterion
        Criterion=(AccountBalance()-StartBalance)/MaxDrawDown;
        //ファイルへ書き出し
        int h3=FileOpen("opt_data.txt",FILE_CSV | FILE_WRITE | FILE_READ,');
        FileSeek(h3, 0, SEEK_END);
        FileWrite(h3,IZYOU+"¥t¥t",IKA+"¥t¥t",Criterion);
        FileClose(h3);
    }
    //
    return(0);
}
////////////////////////////////////
int start()
{
    //=====
    // 「任意指標の最適化」 実施用
    if(IsTesting() && IsOptimization())
    {
        if(AccountEquity()>MaxEquity)
        {
            MaxEquity=AccountEquity();

```

```
    }
    //
    if((MaxEquity-AccountEquity())>MaxDrawDown)
    {
        MaxDrawDown=MaxEquity-AccountEquity();
    }
}
//=====

if(Bars<100 || IsTradeAllowed()==false) return;
// NewBar かチェック
if(IsNewBar() && (Bars>barsTotal))
{
    barsTotal=Bars;

    //＜ポジションが在る場合の処理＞
    if(OrdersTotal()>=1)
    {
        for(int i=0;i<OrdersTotal();i++)
        {
            if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==false) break;
            if(OrderMagicNumber()!=Magic_ID || OrderSymbol()!=Symbol()) continue;

            if(OrderMagicNumber()==Magic_ID && OrderSymbol()==Symbol())
            {
                bool ikisugi=(Time[0]-in_time)>=(300*Period()*60);

                if(OrderType()==OP_BUY)
                {
                    double kachi_buy_price=OrderOpenPrice()+profit;
                    double make_buy_price=OrderOpenPrice()-loss;
                    //
                    bool kachi_buy=(Bid>=kachi_buy_price);
                    bool make_buy=(Bid<=make_buy_price);
                    //
                    if(kachi_buy || make_buy || ikisugi)
                    {
                        OrderClose(OrderTicket(),Lots,Bid,0,Blue);
                        continue;
                    }
                }

                if(OrderType()==OP_SELL)
                {
                    double kachi_sell_price=OrderOpenPrice()-profit;
                    double make_sell_price=OrderOpenPrice()+loss;
                    //
                    bool kachi_sell=(Ask<=kachi_sell_price);
                    bool make_sell=(Ask>=make_sell_price);
                    //
                    if(kachi_sell || make_sell || ikisugi)
                    {
                        OrderClose(OrderTicket(),Lots,Ask,0,Blue);
                        continue;
                    }
                }
            }
        }
    }
}
//end_of_if(OrdersTotal()>=1)

//＜新規ポジションの設定と方針＞
//アップトレンドでは⇒buyIN のみ実施（順張り）
//ダウトレンドでは⇒sellIN のみ実施（順張り）
```

```

//レンジ相場では ⇒buy と sell (逆張り?-)
//if(OrdersTotal()<=max_position)
if(OrdersTotal()==0)
{

//判断用の配列を準備
ArraySetAsSeries(Bol_hi,true);
ArraySetAsSeries(Bol_lo,true);
//
for(int j=0;j<=6;j++)
{
    Bol_hi[j]=iBands(NULL,0,period_bollin,2,0,PRICE_OPEN,MODE_UPPER,j); //2_σ
    Bol_lo[j]=iBands(NULL,0,period_bollin,2,0,PRICE_OPEN,MODE_LOWER,j); //2_σ
}

//判断用の共通指標を準備
bool Subtract=((IZYOU<(Bol_hi[0]-Bol_lo[0])) && ((Bol_hi[0]-Bol_lo[0]<IKA));
bool hanareta=((Time[0]-in_time)>=(19*Period()*60));

bool TREND_UP=((slope_kaiki(0,trendPeriod))<=(_up)) && (call_hensa(0,trendPeriod)<=div_);
bool TREND_DOWN=((slope_kaiki(0,trendPeriod))>=(_down)) &&
(call_hensa(0,trendPeriod)<=div_);

////if(TREND_UP)
////{
//USDJPY の上昇相場 (円安)
//買い条件は整ったか
bool cross_up_1=((High[1]>Bol_hi[1]) && (Open[0]>Bol_hi[0]));
bool cross_up_2=((Open[3]<Bol_hi[3]) && (Open[2]<Bol_hi[2]));
bool cross_up_3=((Open[4]<Bol_hi[4]) && (Open[3]<Bol_hi[3]));
bool cross_up_4=((Open[5]<Bol_hi[5]) && (Open[4]<Bol_hi[4]));
bool cross_up_5=((Open[6]<Bol_hi[6]) && (Open[5]<Bol_hi[5]));
bool cross_up=(cross_up_1 && (cross_up_2 || cross_up_3 || cross_up_4 || cross_up_5));
//条件の補足
bool cross_up_rapid=((High[2]>Bol_hi[2]) && (High[1]>Bol_hi[1]) && (Open[0]>Bol_hi[0]));
//Filter ; 本当に買いか
double slope_s_buy=slope_kaiki(0,shortPeriod_buy);
double slope_m_buy=slope_kaiki(0,mediumPeriod_buy);
bool SLOPE_BUY=((slope_s_buy<=short_buy) && (slope_m_buy<=long_buy));
//
if(((cross_up || cross_up_rapid) && SLOPE_BUY) && Subtract && hanareta)
{
    OrderSend(Symbol0,OP_BUY,Lots,Ask,0,0,0,"my_Buy_order",Magic_ID,Green);
    in_time=Time[0];
}

////} else if(TREND_DOWN)
////{
////+
//USDJPY の下降相場 (円高)
//売り条件は整ったか
bool cross_down_1=((Low[1]<Bol_lo[1]) && (Open[0]<Bol_lo[0]));
bool cross_down_2=((Open[3]>Bol_lo[3]) && (Open[2]>Bol_lo[2]));
bool cross_down_3=((Open[4]>Bol_lo[4]) && (Open[3]>Bol_lo[3]));
bool cross_down_4=((Open[5]>Bol_lo[5]) && (Open[4]>Bol_lo[4]));
bool cross_down_5=((Open[6]>Bol_lo[6]) && (Open[5]>Bol_lo[5]));
bool cross_down=(cross_down_1 && (cross_down_2 || cross_down_3 || cross_down_4 ||
cross_down_5));
//条件の補足
bool cross_down_rapid=((Low[2]<Bol_lo[2]) && (Low[1]<Bol_lo[1]) &&
(Open[0]<Bol_lo[0]));
//Filter ; 本当に売るか
double slope_s_sell=slope_kaiki(0,shortPeriod_sell);
double slope_m_sell=slope_kaiki(0,mediumPeriod_sell);
bool SLOPE_SELL=((slope_s_sell>=short_sell) && (slope_m_sell>=long_sell));

```



```

//
if(((cross_down || cross_down_rapid) && SLOPE_SELL) && Subtract && hanareta)
{
    OrderSend(Symbol(),OP_SELL,Lots,Bid,0,0,0,"my_Sell_order",Magic_ID,Red);
    in_time=Time[0];
}
/**/
////} else if(TREND_UP==false && TREND_DOWN==false)
////{
//レンジ相場
//内容は検討中です
////}

} //end_of_if(OrdersTotal()<=max_position)

} else
{
// [5分足] 内で起こる急変に対応する処理を記載する
// バックテストでは確認が困難な部分

if(OrdersTotal()>=1)
{
for(int ii=0;ii<OrdersTotal();ii++)
{
if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==false) break;
if(OrderMagicNumber()!=Magic_ID || OrderSymbol()!=Symbol()) continue;

if(OrderMagicNumber()==Magic_ID && OrderSymbol()==Symbol())
{

if(OrderType()==OP_BUY)
{
double i_kachi_buy_price=OrderOpenPrice()+profit_2;
double i_make_buy_price=OrderOpenPrice()-loss_2;
//
bool i_kachi_buy=(Bid>=i_kachi_buy_price);
bool i_make_buy=(Bid<=i_make_buy_price);
//
if(i_kachi_buy || i_make_buy)
{
OrderClose(OrderTicket(),Lots,Bid,0,Blue);
continue;
}
}

if(OrderType()==OP_SELL)
{
double i_kachi_sell_price=OrderOpenPrice()-profit_2;
double i_make_sell_price=OrderOpenPrice()+loss_2;
//
bool i_kachi_sell=(Ask<=i_kachi_sell_price);
bool i_make_sell=(Ask>=i_make_sell_price);
//
if(i_kachi_sell || i_make_sell)
{
OrderClose(OrderTicket(),Lots,Ask,0,Blue);
continue;
}
}

}
}
}
} //end_of_if(OrdersTotal()>=1)

```

```

    }

    return(0);
}
////////// 以下は関数類 //////////
bool IsNewBar()
{
    datetime curbar = Time[0]; // Open time of current bar
    if (lastbar!=curbar)
    {
        lastbar=curbar;
        return (true);
    }
    return(false);
}
//-----
double slope_kaiki(int start_s,int period_s)
{
    ArraySetAsSeries(Price_01,true);
    Price_01[0]=Open[0];
    for(int j_=(start_s+1);j_<=((start_s+1)+(2*period_s));j_++)
    {
        Price_01[j_]=(1.0/4.0)*(Open[j_]+High[j_]+Low[j_]+Close[j_]);
    }
    double slope=kaiki_sen(Price_01,start_s,period_s,0);
    //
    return(slope);
}
//
double call_hensa(int start_h,int period_h)
{
    double koubai_M=kaiki_sen(Price_01,start_h,period_h,0);
    double seppen_M=kaiki_sen(Price_01,start_h,period_h,1);
    //
    double ooM=0;
    double sigma=0;
    for(int pM=start_h;pM<=((start_h+period_h));pM++)
    {
        Trend_[pM]=koubai_M*ooM+seppen_M;
        ooM=ooM+1;
        //
        double div_=(Price_01[pM]-Trend_[pM])*(Price_01[pM]-Trend_[pM]);
        sigma=sigma+div_;
    }
    double div=sigma/100;//100 足あたりの値
    //Print 内容は、log ファイルにも記録されるので、バックテスト時に異常解析の邪魔になる
    // Print("勾配は= ",koubai_M);
    // Print("回帰線からの偏差= ",div);
    // Comment("勾配は= ",koubai_M,"回帰線からの偏差= ",div);
    //---
    return(div);
}
//-----
//回帰直線の傾き、切辺を返す
double kaiki_sen(double& price[],int start_,int period_,int what_)
{
    double X_av=0.0,X_i=0.0;
    double Y_av=0.0,Y_i=0.0;
    double i_D;
    //
    ArraySetAsSeries(price,true);
    //まず、平均値を求める
    for(int i=start_;i<=((start_+period_));i++)
    {
        i_D=i*1.0;

```

```

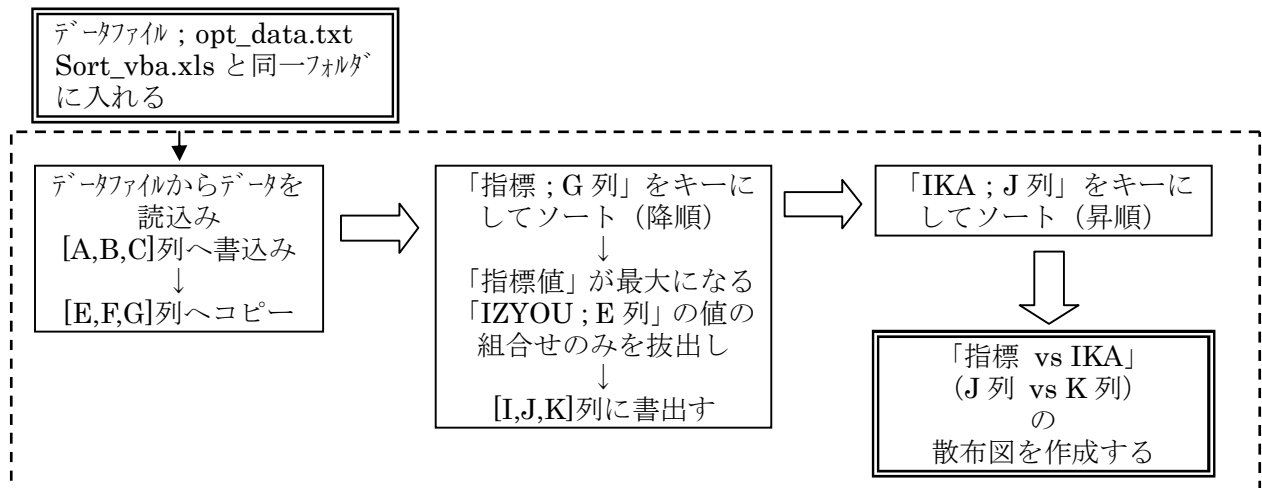
    X_i=X_i+i_D;
    Y_i=Y_i+price_[i];
}
X_av=(X_i/(period_+1));
//X_av=(start_+(start_+period_))/period_;
Y_av=(Y_i/(period_+1));
//加算
double bunshi=0.0,bunbo=0.0;
double koubai_=0.0,seppen_=0.0;
double j_D;
//
for(int j=start_;j<=(start_+period_);j++)
{
    j_D=j*1.0;
    bunshi=bunshi+(j_D-X_av)*(price_[j]-Y_av);
    bunbo=bunbo+(j_D-X_av)*(j_D-X_av);
}
koubai_=bunshi/bunbo;
/////seppen_=Y_av-(koubai_*X_av);//良い値が出ない
seppen_=price_[start_];
//
if(what_==0) return(koubai_);
if(what_==1) return(seppen_);
//
}
//*****

```

4. エクセルによるデータ処理例

(1) 処理フロー概要

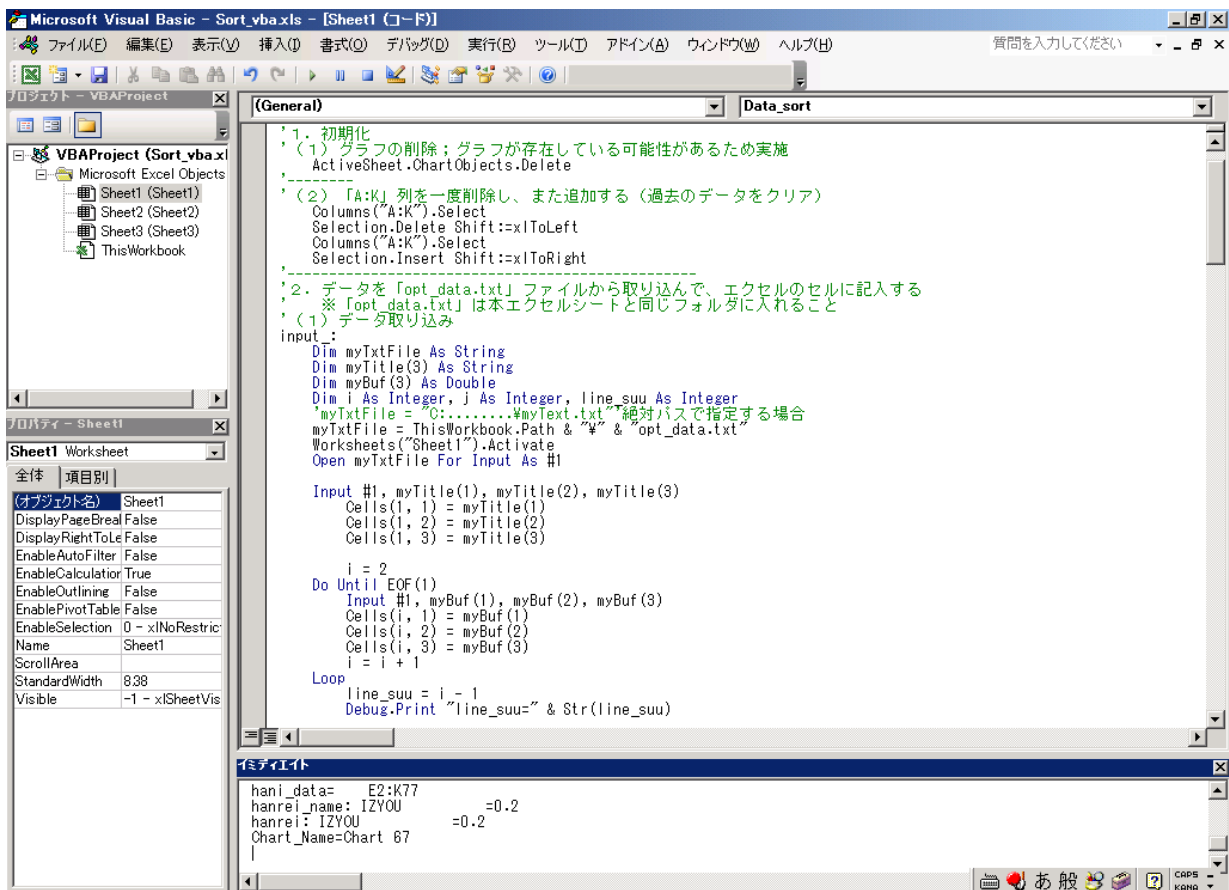
- ・エクセルのマクロ名称；
「Sort_vba.xls」ファイル中の、「Sheet1」の「(general)」に設定した、「Data_sort」
- ・「Data_sort」は、下記の手順（破線で囲んだ部分）を実行します。



(2) 実施例

①実行方法

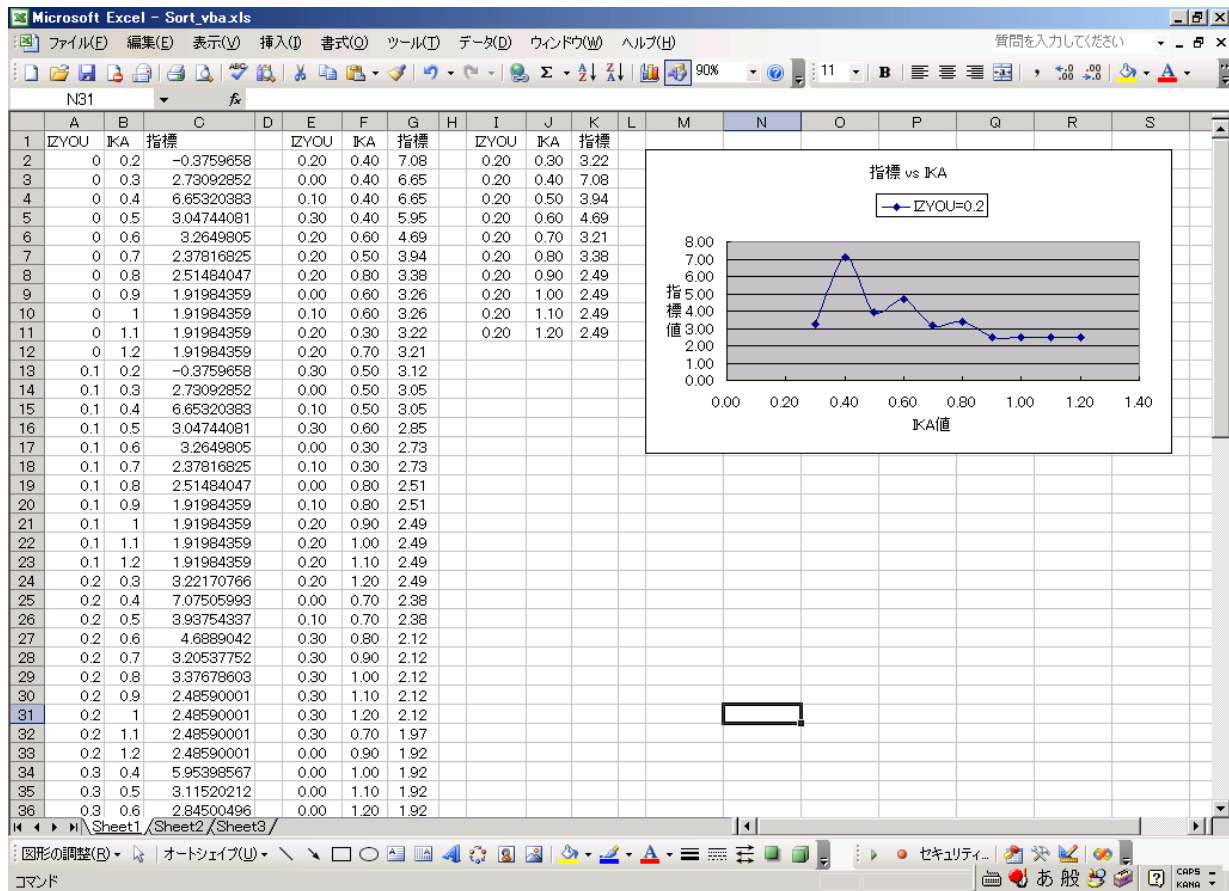
- ・「Sort_vba.xls」を開いて、更に「Visual Basic Editor」を開きます。
- 実行する一方法は、[実行] - [Sub/ユーザーフォームの実行] を選択します。(F5)でも OK)



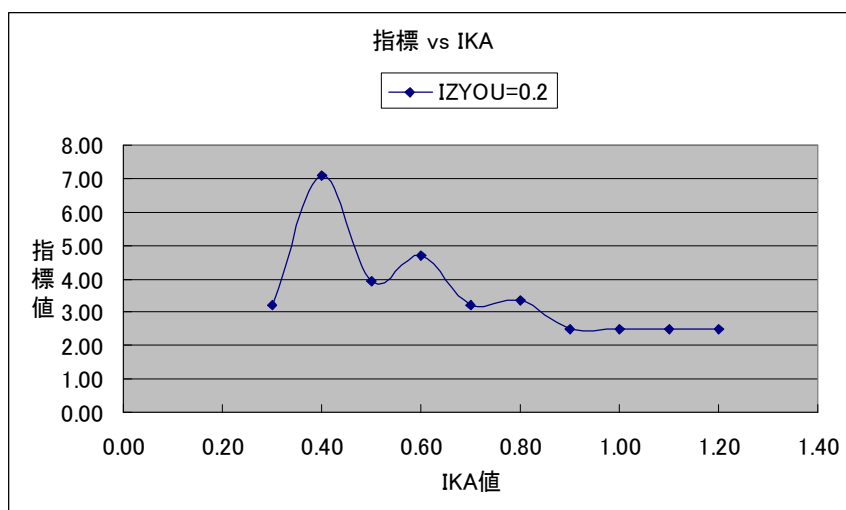
※ [表示] - [イミディエイト ウィンドウ] とすると、デバッグ用の窓も開きます。

②実行した後の「Sheet1」例

- ・実行し、エラーがなければエクセル（Sheet1）に戻ってください、
下記の様な結果が得られます。



グラフ部の拡大；



③結果まとめ

- ・パラメータ「IKA=0.4、IZYOU=0.2」が最適値と言う結果は、「資産（Balance）」の最適化と同一である。（ちょっと、つまらなかった！）
 - ・「IZYOU=0.2（一定）」としたとき、最適値（IKA=0.4）が不安定な位置にあることに、気が付いた！（Balanceの最適化の場合はどうなっているのか気がかり！）
- 本来は「IKA」を少々動かしても、「指標値」は余り変化しない「安定点」が望ましい。

(3) マクロ・コード (VBA)

- ・「Sort_vba.xls」ファイル中の、「Sheet1」の「(general)」に設定した、「Data_sort」です
- ・一見難しそうですが、実は下記部分のコードは、殆ど「マクロの記録」で自動作成することが出来ます。

※マクロを動かすには、必ず「tester/files」フォルダから「opt_data.txt」を、「Sort_vba.xls」の在るフォルダにコピーしてから実施してください。

下記のマクロでは、「opt_data.txt」は同一のフォルダにあるものとして、記述しました。

①「マクロの記録」で作成可能部分

下記コードの内、

1. 初期化
3. データのソート（並べ替え）を行う
4. グラフ化の準備
5. グラフ化

※「マクロの記録」結果がそのままでは使えなくとも、若干の修正で正常に動作するマクロが作れてしまいます。（ネット等で調べれば、「マクロの記録」の使い方は簡単に判ります）

※「2. データを「opt_data.txt」ファイルから取り込んで、エクセルのセルに記入する」のみが、資料を参照しながらアメンボが独自に考えた部分です。

②コード内容

'最適化「指標」のデータを取り込み、グラフを作作成する

Sub Data_sort()

' 1. 初期化

' (1) グラフの削除；グラフが存在している可能性があるため実施
ActiveSheet.ChartObjects.Delete

' (2) 「A:K」列を一度削除し、また追加する（過去のデータをクリア）

Columns("A:K").Select

Selection.Delete Shift:=xlToLeft

Columns("A:K").Select

Selection.Insert Shift:=xlToRight

' 2. データを「opt_data.txt」ファイルから取り込んで、エクセルのセルに記入する

' ※「opt_data.txt」は本エクセルシートと同じフォルダに入れること

' (1) データ取り込み

input_:

Dim myTxtFile As String

Dim myTitle(3) As String

Dim myBuf(3) As Double

Dim i As Integer, j As Integer, line_suu As Integer

'myTxtFile = "C:.....¥myText.txt"絶対パスで指定する場合

myTxtFile = ThisWorkbook.Path & "¥" & "opt_data.txt"

Worksheets("Sheet1").Activate

Open myTxtFile For Input As #1

Input #1, myTitle(1), myTitle(2), myTitle(3)

```

Cells(1, 1) = myTitle(1)
Cells(1, 2) = myTitle(2)
Cells(1, 3) = myTitle(3)

i = 2
Do Until EOF(1)
    Input #1, myBuf(1), myBuf(2), myBuf(3)
    Cells(i, 1) = myBuf(1)
    Cells(i, 2) = myBuf(2)
    Cells(i, 3) = myBuf(3)
    i = i + 1
Loop
line_suu = i - 1
Debug.Print "line_suu=" & Str(line_suu)
Close #1
'-----
' (2) 元データは保存し、コピーしたデータを加工用として準備する
Dim p As Integer
For p = 1 To line_suu
    Cells(p, 5).Value = Cells(p, 1).Value
    Cells(p, 6).Value = Cells(p, 2).Value
    Cells(p, 7).Value = Cells(p, 3).Value
Next
'-----
' 3. データのソート（並べ替え）を行う
sort_:
Dim hani As String
hani = "E1:G" & line_suu
Debug.Print "hani=" & hani

Worksheets("Sheet1").Activate
Range(hani).Sort _
    key1:=Range("G1"), _
    Order1:=xlDescending, _
    Header:=xlYes, _
    Orientation:=xlTopToBottom
'-----
' 4. グラフ化の準備
' (1) 「指標値」が最大に成る「パラメータ」の組合せのうち、
'     「I列値＝一定」のもののみを抜き出す。
graph_:
Dim k As Integer, m As Integer, n As Integer
Dim opt As Double
Dim g_line_suu As Integer

Cells(1, 9).Value = Cells(1, 5).Value
Cells(1, 10).Value = Cells(1, 6).Value
Cells(1, 11).Value = Cells(1, 7).Value

opt = Cells(2, 5).Value

n = 2
For k = 2 To line_suu
    If Cells(k, 5).Value = opt Then
        Cells(n, 9).Value = Cells(k, 5).Value
        Cells(n, 10).Value = Cells(k, 6).Value
        Cells(n, 11).Value = Cells(k, 7).Value
        n = n + 1
    End If

```

```

Next
g_line_suu = n - 2
Debug.Print "g_line_suu=" & g_line_suu
'-----
' (2) 抜き出したデータを、更に「J列値」をキーとして並べ替える
re_sort_:
    Dim hani_g As String
    hani_g = "J1:K" & g_line_suu
    Debug.Print "hani_g=" & hani_g

    Worksheets("Sheet1").Activate
    Range(hani_g).Sort _
        key1:=Range("J1"), _
        Order1:=xlAscending, _
        Header:=xlYes, _
        Orientation:=xlTopToBottom
'-----
' (3) データを整形する
arrange_:

    Dim hani_all As String, hani_data As String
    hani_all = "E1:k" & line_suu
    hani_data = "E2:K" & line_suu

    Range(hani_all).Select
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
    End With

    Range(hani_data).Select
    Debug.Print "hani_data=", hani_data
    With Selection
        .NumberFormatLocal = "0.00"
    End With
'-----
' (4) セル幅を整形する
Columns("A:A").EntireColumn.AutoFit
Columns("B:B").EntireColumn.AutoFit
Columns("C:C").EntireColumn.AutoFit
Columns("D:D").ColumnWidth = 2.5
Columns("E:E").EntireColumn.AutoFit
Columns("F:F").EntireColumn.AutoFit
Columns("G:G").EntireColumn.AutoFit
Columns("H:H").ColumnWidth = 2.5
Columns("I:I").EntireColumn.AutoFit
Columns("J:J").EntireColumn.AutoFit
Columns("K:K").EntireColumn.AutoFit
Columns("L:L").ColumnWidth = 2.5
'-----
'5. グラフ化
' (1) グラフ作成
graph2_:

    Dim hanrei_name As String
    Dim hanrei_data As String
    Dim hanrei As String

    hanrei_name = CStr(Cells(1, 9).Value)

```



```

hanrei_data = CStr(Cells(2, 9).Value)
hanrei = hanrei_name & "=" & hanrei_data

Debug.Print "hanrei_name: " & hanrei_name & "=" & hanrei_data
Debug.Print "hanrei: "; hanrei

Range("J1:K11").Select
Charts.Add
ActiveChart.ChartType = xlXYScatterSmooth
ActiveChart.SetSourceData Source:=Worksheets("Sheet1").Range("J1:K11"), PlotBy _
    :=xlColumns
'ActiveChart.SeriesCollection(1).Name = ""IZYOU=0.2""
ActiveChart.SeriesCollection(1).Name = hanrei
ActiveChart.Location Where:=xlLocationAsObject, Name:="Sheet1"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "指標 vs IKA"
    .ChartTitle.Characters.Text = Cells(1, 11).Value & " vs " & Cells(1, 10).Value
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "IKA 値"
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = Cells(1, 10).Value & "値"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "指標値"
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = Cells(1, 11).Value & "値"
    '---- Y軸のタイトルを縦書きに変更
    .Axes(xlValue, xlPrimary).AxisTitle.Orientation = xlVertical
End With

' 「凡例」 の表示と位置の設定
ActiveChart.HasLegend = True
ActiveChart.Legend.Select
Selection.Position = xlTop
ActiveChart.Axes(xlValue).AxisTitle.Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .ReadingOrder = xlContext
    .Orientation = xlVertical
End With
'-----
' (2) チャート (グラフ) をデータと重ならない場所に移動する
Dim name_ As String
Dim index_ As Integer

name_ = ActiveChart.Parent.Name
Debug.Print "Chart_Name=" & name_

With ActiveSheet.ChartObjects(name_)
    .Top = Range("M2").Top
    .Left = Range("M2").Left
End With
'
End_
End Sub
'++++++

```

以上