

高頻度データに対する 多目的効用最大化のための ストラテジーの自動チューニング

**Automatic Tuning of Strategies
for Multiobjective Utility Maximization
for High Frequency Data.**

大谷 和樹

富山県立大学 情報基盤工学講座
t515010@st.pu-toyama.ac.jp

**Teams, 9:50-10:15 Friday, December 4, 2020,
Toyouma Prefectural University.**

はじめに
高頻度データ収集
直交表にもとづく
ロバスト設計
提案手法
実験結果ならびに
考察
おわりに

1.1 本研究の背景 1

2/19

背景

経済活動の活発化に伴い、金融市場の規模は拡大し金融市場のメカニズムに関する研究の重要性が増している。しかし、金融市場は複雑な要因が絡み合っているため全容を明らかにすることは困難である。

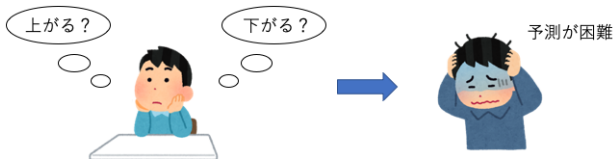


図 1: 金融市場の予測

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに

1.1 本研究の背景 2

3/19

Long Term Capital Management(LTCM)

ソロモン・ブラザーズ出身のジョン・メリウェザーが設立したファンドで、ブラック・ショールズ方程式を完成させ 1997 年にノーベル経済学賞を受賞したマイロン・ショールズとロバート・マートンもメンバーだった。最盛期には 1,000 億ドル程度の資産規模まで拡大していたが、1988 年のロシア危機が引き金となり 1 か月で純資産の半分を失い破産に追い込まれる¹。

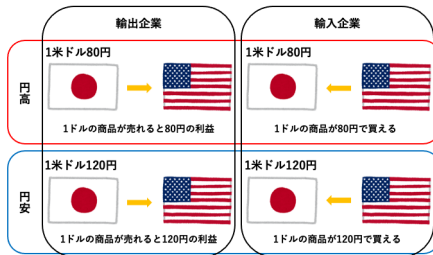


図 2: 円高円安と企業の関係

¹ 旦 直土, “銀行員のための教科書, 「LTCM から得る教訓の重要性」”,
[https://www.financepensionrealestate.work/entry/2017/09/03/111854\(2020/12/25\)](https://www.financepensionrealestate.work/entry/2017/09/03/111854(2020/12/25)) .

1.2 本研究の目的

4/19

目的

過去の研究の多くは複数ルールは適用していたとしても、効率的にルールを選ぶ方法は組み込まれていない。そこで、本研究ではリアルタイムで取得した Tick データから指標を算出し、より最適な売買ルールを自動的に選択する金融マーケット予測手法の開発を目的とした分析手法を考える。

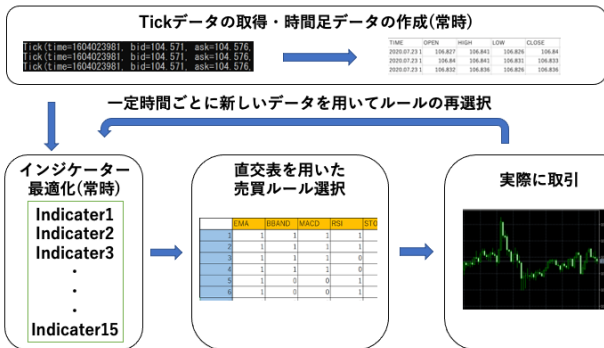


図 3: 提案手法の流れ

2.1 取引プラットフォーム

5/19

MetaTrader 5(MT5)

- 外国為替及び為替市場におけるテクニカル分析及び取引業務を行うトレーダー向けの無料アプリケーションである。
- Python を使って MT5 から Tick の情報を取得したり売買の指令を送ることができるため、取得した Tick の情報を利用した自動売買を行うことができる。

```
Tick(time=1604023981, bid=104.571, ask=104.576, last=123.6,
Tick(time=1604023981, bid=104.571, ask=104.576, last=123.6,
Tick(time=1604023981, bid=104.571, ask=104.576, last=123.6,
Tick(time=1604023981, bid=104.571, ask=104.576, last=123.6,
```



MT5から高頻度データを取得→Pythonで分析し値動きを予測
→PythonでMT5に売買オーダーを送信→MT5で売買処理

図 4: MT5 と Python の関係

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに

2.2 インジケーターを用いたテクニカル分析 1

6/19

テクニカル分析

過去と現在の価格や出来高などのデータからテクニカル指標を算出することによって傾向を把握し、これからの値動きについて未来の価格の予測を行う。

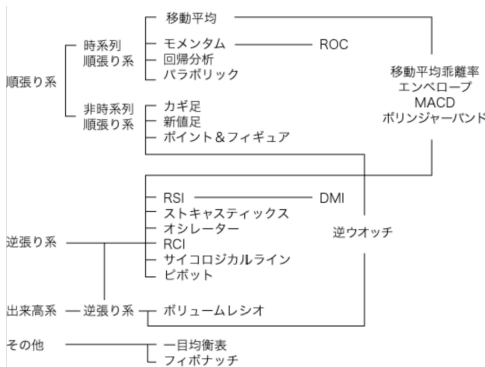


図 5: テクニカル分析に使われる指標

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに

2.2 インジケーターを用いたテクニカル分析 2

7/19

インジケーター

テクニカル分析に使用するために、為替レートの時系列情報を様々な計算で加工して売買の判定に利用する指標のことを指す。
特定期間内の平均レートや相場の方向感を数値化したデータなどが例として挙げられる。

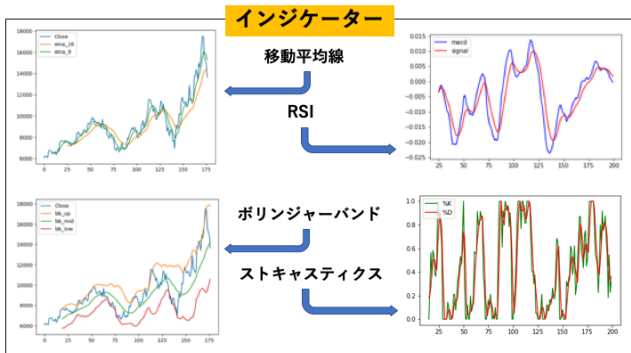


図 6: インジケーターの例

2.3 ストラテジーテスターにおける バックテストと最適化 1



8/19


ストラテジーテスター

MT5 内の売買ルールに値するエキスパートアドバイザーの最適化を行うためのプログラム

backtesting.py

オープンソースとして公開されているバックテストのライブラリ。
特徴：処理が高速・TA-Lib をサポートしている

エキスパート: Examples\Moving Average\Moving Average ex5  

銘柄: USDJPY MT 

日付: 先月 2020.12.01 2020.12.24

フォワードテスト: キャンセル 2020.06.14


遅延: 遅延ゼロ, 理想的な実行  取引実行中にスリッページとリクオートをエミュレートする遅延を選択します

図 7: ストラテジーテスター設定画面

```
#EMA最適化
print("-----EMA-----")
bt = Backtest(df_, EMAt, cash=10000, commission=.0012, trade_on_close=True)
m=list(range(10, 50, 5))
n=list(range(15, 100, 5))
r=list(range(10, 30, 5))
stats = bt.optimize(m=m,n=n,r=r,maximize='Equity Final [$]',constraint=lambda p: p.m < p.n)
```

図 8: Backtesting.py で最適化を行う部分のコード

はじめに
高頻度データ収集
直交表にもとづく
ロバスト設計
提案手法
実験結果ならびに
考察
おわりに

2.3 ストラテジーテスターにおける バックテストと最適化 2

9/19

```
-----EMA-----
Start                2021-01-28 14:25:10
End                  2021-01-29 04:15:20
Duration              0 days 13:50:10
Exposure Time [%]    97.511
Equity Final [$]     100192
Equity Peak [$]      100229
Return [%]           0.192487
Buy & Hold Return [%] -0.0594542
Calmar Ratio         99.1375
Max. Drawdown [%]   -0.0498184
Avg. Drawdown [%]   -0.00834093
Max. Drawdown Duration 0 days 08:44:00
Avg. Drawdown Duration 0 days 00:30:45
# Trades              6
Win Rate [%]         66.6667
Best Trade [%]        0.087142
Worst Trade [%]       -0.0162675
Avg. Trade [%]        0.0446741
Max. Trade Duration   0 days 09:09:30
Avg. Trade Duration   0 days 03:43:50
Profit Factor         10.0028
Expectancy [%]        0.0446835
SQN                   1.64307
strategy              EMAt (m=6, n=20, r=...
```

- ・ヒストリカルデータの開始日時
- ・ヒストリカルデータの終了日時
- ・ヒストリカルデータの期間
- ・ポジションを持っていた期間の割合
- ・所持金の最終値
- ・所持金の最高値
- ・利益率=損益÷開始時所持金×100
- ・((終了時の終値 - 開始時の終値) ÷ 開始時の終値)の絶対値×100
- ・最大損失率に対する年間平均収益の比率
- ・最大下落率
- ・平均下落率
- ・最大下落期間
- ・平均下落期間
- ・取引回数
- ・勝率=勝ち取引回数÷全取引回数×100
- ・1回の取引での利益の最大値÷所持金×100
- ・1回の取引での損失の最大値÷所持金×100
- ・損益の平均値÷所持金×100
- ・1回の取引での最長期間
- ・1回の取引での平均期間
- ・総利益と総損失の比率=総利益÷総損失
- ・期待値=平均利益×勝率+平均損失×敗率
- ・SQN (System Quality Number)
- ・最適化の結果

図 9: Backtesting.py で最適化した結果

3.1 ロバストパラメータ設計

10/19

ロバスト設計

様々なノイズ（誤差因子）に強い設計を行う強力な設計最適化手法。
ばらつきの原因となる誤差因子をコントロールするのではなく，設計に有効な制御因子と誤差因子の交互作用（誤差因子の影響がなるべく小さくなるような制御因子の水準条件）を見つけることにより誤差因子の影響を減衰させようとするものである²。

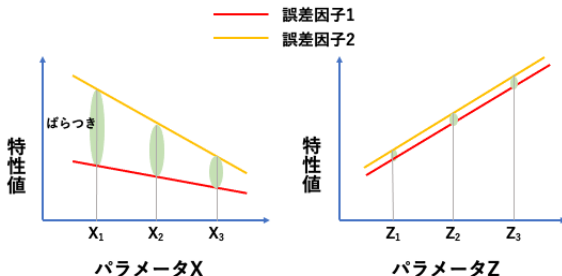


図 10: ロバスト設計のイメージ図

²金井喜美雄，“ロバスト適応制御入門”，寺尾満監修，オーム社，1989.

3.2 直交表と実験計画法 1

11/19

直交表

すべての列において数値の組み合わせが均衡，列同士の相関が全くない，列同士の成分を掛け合わせた値を持つ列が存在するといった性質を持っている。

直交表の列に要因を割り付け，それぞれの水準を対応させることで調査条件を設定することで計画行列と見ることができる³。

直交表 $L_8(2^7)$

No.	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

- ・ 2 水準の要因を 7 個まで扱える
- ・ 多元配置の $128(2^7)$ 通りの組み合わせが 8 通りの組み合わせで確認できる

直交表 $L_9(3^4)$

No.	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

- ・ 3 水準の要因を 4 個まで扱える
- ・ 多元配置の $81(3^4)$ 通りの組み合わせが 9 通りの組み合わせで確認できる

図 11: 直交表の例

³ 淵上美喜ほか，“Excel で学ぶ営業・企画・マーケティングのための実験計画法”，オーム社，2006。

3.2 直交表と実験計画法 2

12/19

実験計画法

直交表を用いた要因計画は、実験計画法のうちの一つである。
実験計画法では、直交表を回帰分析し、その測定結果 y_{ijk} をモデル化する (i, j, k はパラメータの水準を表す)。
これにより、測定結果における各要因の主効果を求めることができる。⁴

No.	ルール1	ルール2	ルール3				結果
1	1	1	1	1	1	1	4.4
2	1	1	1	0	0	0	-5.7
3	1	0	0	1	1	0	8.1
4	1	0	0	0	0	1	4.6
5	0	1	0	1	0	1	-3.5
6	0	1	0	0	1	0	7.2
7	0	0	1	1	0	0	2.1
8	0	0	1	0	1	1	-8.9

パラメータが3つの場合のモデル式

$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\beta\gamma)_{jk} + (\alpha\gamma)_{ik} + (\alpha\beta\gamma)_{ijk} + \varepsilon_{ijk}$$

上記のような
直交表を
分散分析

μ : パラメータによらない成分
 $\alpha_i, \beta_j, \gamma_k$: パラメータA, Bそれぞれによる効果
 $(\alpha\beta\gamma)_{ijk}$: パラメータ間の交互作用による効果
 ε_{ijk} : 測定の誤差

図 12: モデル式

⁴ 杉本章義, 加藤和彦, “実験計画法を利用したウェブサーバの主要なパラメータ選択手法”, 情報処理学会 研究報告 pp. 33-40 2008.

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに

4.1 データ取得方法

13/19

繰り返し

MT5からTickデータを取得

```
Tick(time=1604023981, bid=104.571, ask=104.576)
Tick(time=1604023981, bid=104.571, ask=104.576)
Tick(time=1604023981, bid=104.571, ask=104.576)
Tick(time=1604023981, bid=104.571, ask=104.576)
Tick(time=1604023981, bid=104.571, ask=104.576)
Tick(time=1604023981, bid=104.571, ask=104.576)
```

取得時の秒が変わるごとに保存

```

time      price volume
0  2021-01-07 08:37:01 103.0420 0
1  2021-01-07 08:37:04 103.0425 0
2  2021-01-07 08:37:06 103.0420 0
3  2021-01-07 08:37:09 103.0415 0
4  2021-01-07 08:37:10 103.0415 0
5  2021-01-07 08:37:13 103.0415 0

```

指定秒足ごとにOHLCVの形にリサンプリング

TIME	OPEN	HIGH	LOW	CLOSE
2020.07.23 1	106.827	106.841	106.826	106.84
2020.07.23 1	106.84	106.841	106.831	106.833
2020.07.23 1	106.832	106.836	106.826	106.836

10S

30S

1M

5M

30M

4H

図 13: データ取得の流れ

4.2 パラメータの最適化

14/19

はじめに

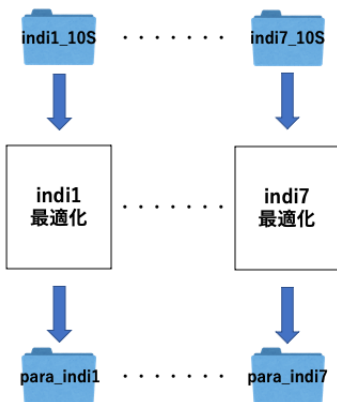
高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに



```

class EMA(Strategy):
    m = 12
    n = 20
    r = 14
    a = 10

    def init(self):
        self.ema1 = self.I(EMA, self.data['Close'], self.m)
        self.ema2 = self.I(EMA, self.data['Close'], self.n)
        self.atr = self.I(ATR, self.data['High'], self.data['Low'], self.data['Close'], self.r)

    def next(self):
        if not self.atr == 0:
            if (self.ema1[2] < self.ema2[2] and self.ema1[1] > self.ema2[1]):
                self.buy(limit=self.data['Close'], tp = self.data['Close'] + (self.atr * self.a), sl = self.data['Close'] - (self.atr * self.a))
            if (self.ema1[2] > self.ema2[2] and self.ema1[1] < self.ema2[1]):
                self.sell(limit=self.data['Close'], tp = self.data['Close'] - (self.atr * self.a), sl = self.data['Close'] + (self.atr * self.a))

df = df.tail(1000)
print(df)
dt_row = datetime.datetime.now()
print(dt_row)
print("-----EMA-----")

bt = Backtest(df.tail(10000), EMA, cash=100000, commission=spread, trade_on_close=True)
m=list(range(6, 18, 2))
n=list(range(10, 32, 2))
r=list(range(6, 24, 2))
a=list(range(6, 21, 2))
stats = bt.optimize(m=m,n=n,r=r,a=a,maximize='Equity Final [$]',constraint=lambda p: p.m < p.n)
  
```

- ・ 売買に使用するインジケーターの計算に使用する期間
- ・ ATRの計算に使用する期間
- ・ ATRの倍率

図 14: パラメータ最適化の流れ

4.3 複数ルールからの最適な組み合わせ選択

15/19

はじめに
高頻度データ収集
直交表にもとづく
ロバスト設計
提案手法
実験結果ならびに
考察
おわりに

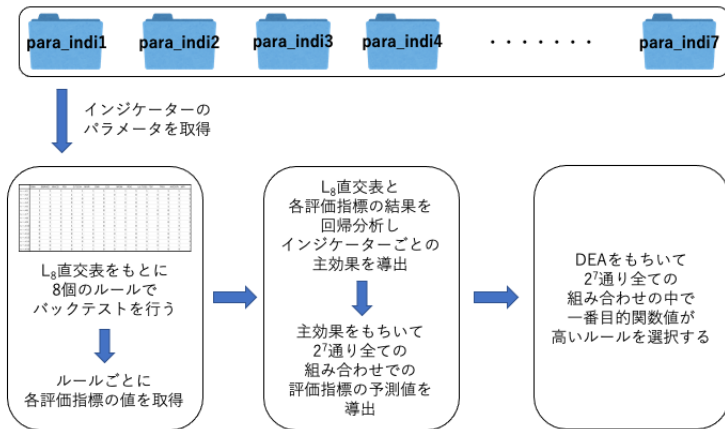


図 15: ルール選択の流れ

自動売買

前段階で得られた最適なルールにもとづいて売買を行う。必要な値は以下の 5 つ。

- 1 symbol
- 2 volume
- 3 price
- 4 sl & tp
- 5 deviation

1,2,5 はあらかじめ決まったの値を設定し，3 はそのときの最新の bid と ask の値を，4 はそれぞれのインジケーターごとに最適化された ATR の値を使用する。

5 実験結果

17/19

実験結果ができれば載せます.

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに

6 おわりに

18/19

まとめ

まとめ

今後の課題

課題

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに

今回の進捗

- ① インジケーター 15 → 8
- ② プログラムほぼ完成

今後の課題

- 1 グラフ描画のプログラム作成
- 2 本論・Beamer・概要

はじめに

高頻度データ収集

直交表にもとづく
ロバスト設計

提案手法

実験結果ならびに
考察

おわりに