

# 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

2/21

## 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。

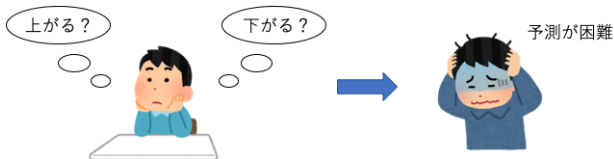


図 1: 金融市場の予測

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

3/21

## Long Term Capital Management(LTCM)

ソロモン・ブラザーズで債券トレーディング部門を率いていたジョン・メリウェザーが設立したファンド。チームメンバーには、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・マートンも加わっていた。

当初は平均の年間利回りが 40 %を突破する等の大成功をおさめ、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<sup>1</sup>。

<sup>1</sup> 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>(参照 2020-12-25).

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



## 2.1 取引プラットフォーム

5/21

### 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, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
tick(time=1604023981, bid=104.571, ask=104.576, last=123.6, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
tick(time=1604023981, bid=104.571, ask=104.576, last=123.6, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
tick(time=1604023981, bid=104.571, ask=104.576, last=123.6, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
tick(time=1604023981, bid=104.571, ask=104.576, last=123.6, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
tick(time=1604023981, bid=104.571, ask=104.576, last=123.6, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
tick(time=1604023981, bid=104.571, ask=104.576, last=123.6, volume=1750000, time_msc=1604023981648, flags=4, volume_real=1750000.0)
```



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 3: MT5 と Python の関係

## 2.2 インジケーターを用いたテクニカル分析

6/21

### 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。

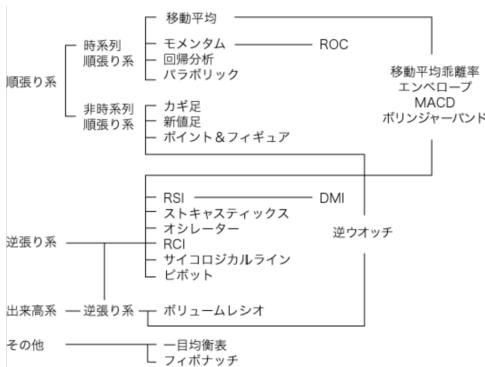


図 4: テクニカル分析に使われる指標

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

7/21

### 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。

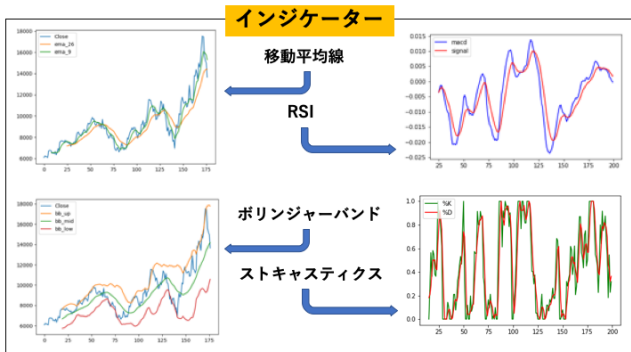


図 5: インジケーターの例

## 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

8/21

### 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

エキスパート: Examples\Moving Average\Moving Average.e5

銘柄: USDJPY 通貨ペア: MT5

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

証券: 選定ゼロ、理想の実行

数字は実行中にスリープ・スキャンをエミュレートする遅延を選択します

実数値	値	スタート	ストップ	ストップ
<input type="checkbox"/> Maximum Risk in percentage	0.02	0.02	0.002	0.2
<input type="checkbox"/> Decrease factor	3	3	0.3	30.0
<input checked="" type="checkbox"/> Moving Average period	12	12	1	120
<input checked="" type="checkbox"/> Moving Average shift	6	6	1	60

PCA	結果	損益	取引数	期待利得	PD-ダウ%	MovingPeriod	MovingShift
0.25	10140.88	140.88	363	0.39	1.10	100	50
0.41	10129.30	129.30	343	0.38	0.61	116	56
0.35	10107.73	107.73	378	0.28	0.90	91	48
0.27	10095.12	95.12	362	0.26	1.01	117	42
11.92	10091.28	91.28	407	0.19	1.63	12	43
0.23	10084.38	84.38	368	0.23	0.82	99	55
0.72	10072.27	72.27	371	0.19	0.99	94	47
0.32	10060.68	60.68	407	0.15	0.81	94	43
0.39	10054.55	54.55	405	0.13	1.27	43	57
0.113	10042.73	42.73	368	0.12	1.29	115	43

図 6: ストラテジーテスターの流れ



## 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

9/21

### 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

```
#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 7: 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

10/21

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

```

Start                2020-09-24 00:00:00
End                  2020-10-01 00:00:00
Duration              7 days 00:00:00
Exposure Time [%]    84.792
Equity Final [$]      10066.9
Equity Peak [$]       10077.6
Return [%]            0.66902
Buy & Hold Return [%] 0.118603
Max. Drawdown [%]    -0.287352
Avg. Drawdown [%]    -0.066725
Max. Drawdown Duration 2 days 00:12:00
Avg. Drawdown Duration 0 days 02:55:00
# Trades              11
Win Rate [%]          90.9091
Best Trade [%]         0.179711
Worst Trade [%]        -0.00645916
Avg. Trade [%]         0.0715777
Max. Trade Duration    2 days 10:09:00
Avg. Trade Duration    1 days 00:07:00
Profit Factor          122.927
Expectancy [%]         0.0727694
SQN                    3.04844
Sharpe Ratio           1.16025
Sortino Ratio           NaN
Calmar Ratio           0.249094
strategy               MACDt (m=10, n=25, ...
equity_curve           ...
trades                 Size ...
  
```

図 8: 最適化した結果

## 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

11/21

### 評価指標

- 1 Exposure Time [%] ポジションを持っていた期間の割合
- 2 Equity Final [\$] 所持金の最終値
- 3 Equity Peak [\$] 所持金の最高値
- 4 Return [%]  $\text{利益率} = \text{損益} \div \text{開始時所持金} \times 100$
- 5 Buy & Hold Return [%]  $((\text{終了時の終値} - \text{開始時の終値}) \div \text{開始時の終値}) \text{の絶対値} \times 100$
- 6 Max. Drawdown [%] 最大下落率
- 7 Avg. Drawdown [%] 平均下落率
- 8 Trades 取引回数
- 9 Win Rate [%]  $\text{勝率} = \text{勝ち取引回数} \div \text{全取引回数} \times 100$
- 10 Best Trade [%]  $1 \text{ 回の取引での利益の最大値} \div \text{所持金} \times 100$
- 11 Worst Trade [%]  $1 \text{ 回の取引での損失の最大値} \div \text{所持金} \times 100$
- 12 Avg. Trade [%]  $\text{損益の平均値} \div \text{所持金} \times 100$

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

## 3.1 ロバストパラメータ設計

12/21

### 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.

パラメータ設計の基本的な考え方は、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.

パラメータ設計は、制御因子の水準変更のみでばらつきの低減を図れるという経済的かつ効果的な方法であるため、ものづくりの設計開発の現場を中心に利用されている<sup>2</sup>.

<sup>2</sup>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

13/21

### 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。

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 9: 直交表の例

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

## 3.2 直交表と実験計画法

14/21

### 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。<sup>3</sup>

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

$\mu$ : 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  
 $\varepsilon_{ijk}$ : 測定の誤差

図 10: モデル式

<sup>3</sup> 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.

## 4.1 データ取得方法

15/21

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)
```

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 11: データ取得の流れ

## 4.1 データ取得方法

16/21

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

	time	price
0	2020/12/11 4:51	104.283
1	2020/12/11 4:51	104.283
2	2020/12/11 4:51	104.283
3	2020/12/11 4:51	104.283
4	2020/12/11 4:51	104.283
5	2020/12/11 4:51	104.283
6	2020/12/11 4:51	104.283
7	2020/12/11 4:51	104.283
8	2020/12/11 4:51	104.283
9	2020/12/11 4:51	104.283
10	2020/12/11 4:51	104.283
11	2020/12/11 4:51	104.283
12	2020/12/11 4:51	104.283
13	2020/12/11 4:51	104.283
14	2020/12/11 4:51	104.283
15	2020/12/11 4:51	104.283
16	2020/12/11 4:51	104.283
17	2020/12/11 4:51	104.283
18	2020/12/11 4:51	104.283
19	2020/12/11 4:51	104.283
20	2020/12/11 4:51	104.283

図 12: Tick データ取得の例

time	open	high	low	close
2020/12/11 4:57	104.2975	104.3005	104.2955	104.3005
2020/12/11 4:57	104.301	104.3015	104.3005	104.3005
2020/12/11 4:57	104.3015	104.3015	104.3005	104.3005
2020/12/11 4:58	104.3005	104.3055	104.3005	104.3055
2020/12/11 4:58	104.305	104.3065	104.3045	104.3055
2020/12/11 4:58	104.3065	104.3065	104.2995	104.3
2020/12/11 4:58	104.3025	104.3025	104.2975	104.2985
2020/12/11 4:58	104.3015	104.3015	104.2995	104.2995
2020/12/11 4:58	104.3	104.301	104.3	104.3005
2020/12/11 4:59	104.301	104.301	104.294	104.2975
2020/12/11 4:59	104.297	104.2975	104.2935	104.2945
2020/12/11 4:59	104.2955	104.2965	104.2945	104.2955
2020/12/11 4:59	104.2945	104.297	104.2945	104.297
2020/12/11 4:59	104.2965	104.2985	104.2965	104.2985
2020/12/11 4:59	104.299	104.3045	104.297	104.304
2020/12/11 5:00	104.3035	104.3325	104.3025	104.325
2020/12/11 5:00	104.3225	104.3265	104.3175	104.3215
2020/12/11 5:00	104.321	104.3345	104.318	104.3295
2020/12/11 5:00	104.329	104.334	104.3245	104.3325
2020/12/11 5:00	104.33	104.3325	104.3225	104.325

図 13: resample 後 (10s)



## 4.2 パラメータの最適化

17/21

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

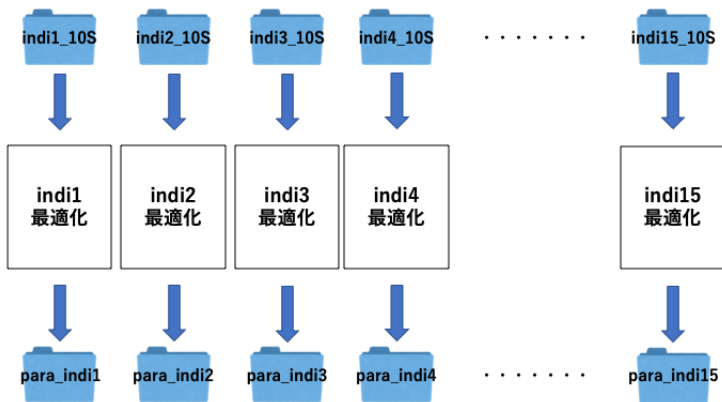


図 14: パラメータ最適化の流れ

## 4.3 複数ルールからの最適な組み合わせ選択

18/21

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

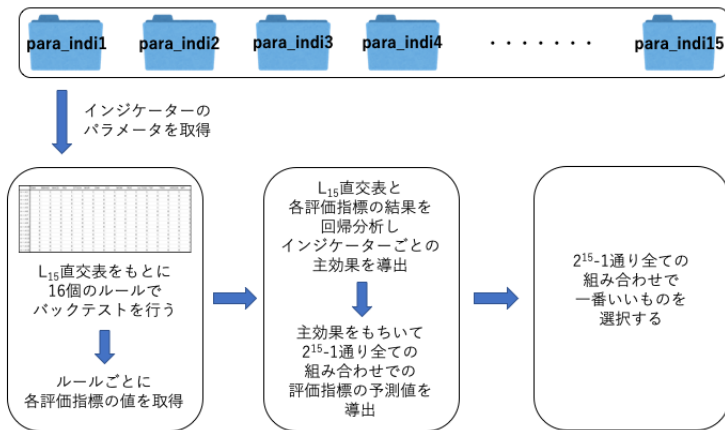


図 15: ルール選択の流れ

## 4.3 複数ルールからの最適な組み合わせ選択

19/21

### 直交表を用いたルール選択

鋭意制作中. システムの流れは

- 1 Tick データを取得し指定した秒足のデータフレームを作成するプログラム (完成)
- 2 OHLCV データからインジケータの最適化を行うプログラム (完成)
- 3 最適化したインジケータのパラメータを使用して直交表にもとづいたルールのバックテストを行い, 主効果を算出 (完成)
- 4 主効果を用いて  $2^{15} - 1$  通りの組み合わせの評価指標の予測値を算出 (完成)
- 5 DEA をもちいて最適なルールを見つける (まだ)
- 6 実際に売買を行うプログラム (まだ)

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

## 5 実験結果

20/21

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

### 今回の進捗

- ① Tick を取得し OHLCV にリサンプルするプログラムの改良
- ② インジケータの計算のプログラムの改良
- ③ ルールごとにバックテストを行い、回帰分析により主効果を算出、それらを用いて  $2^{15} - 1$  通りの組み合わせの評価指標の予測値を算出する部分まで完成

### 今後の課題

- 1 DEA を用いた最適なルール探索の部分
- 2 自動売買をするプログラムの作成
- 3 本論書く