

Nuorium Optimizer V27 Excel アドインマニュアル



株式会社NTTデータ数理システム

Nuorium Optimization Team | NTTDATA MSI

目次

1	はじめに	5
1.1	Excel アドインとは	5
1.2	ご利用になる前に	6
1.2.1	Python 環境	6
1.2.2	互換性について	7
2	チュートリアル	9
2.1	例題の紹介	9
2.2	モデルの記述	10
2.3	モデルを開く	12
2.4	データの登録	13
2.5	実行	16
2.6	結果の出力	17
3	最適化モデル	19
3.1	モデルの書き方	19
3.1.1	PySIMPLE	19
3.1.2	C++SIMPLE	20
3.2	モデルを開く	20
3.3	モデルの編集	22
3.4	モデルの更新	22
4	データの登録	23
4.1	範囲を登録	23
4.2	範囲の詳細設定	24
4.3	自動付番	26
4.4	外部入力ファイル	26
5	実行	29
5.1	実行	29
5.2	停止	31

6	結果の出力	33
6.1	新規シートに出力	33
6.2	範囲に出力	33
6.3	範囲の詳細設定	35
6.4	自動付番	36
6.5	結果のクリア	36
6.6	結果の取込	37
6.7	solve インデックス	38
6.8	下限値／上限値／双対値	39

Last Update 2025 年 03 月 31 日

はじめに

このドキュメントでは Nuorium Optimizer Excel アドインの基本的な使い方を説明します。最初にチュートリアルで一通りの流れを紹介し、次に各機能の使用方法を説明します。

1.1 Excel アドインとは

Excel アドインは数値最適化パッケージ Nuorium Optimizer の表計算ソフト Microsoft Excel 用のインターフェースです。最適化モデルと Excel 上のデータとの橋渡し役を担います。

Excel アドインの特徴を以下に挙げます。

手軽に最適化

モ

プログラミング言語 PySIMPLE/C++SIMPLE で書かれたモデルファイルを選択し、Excel シート上のデータを登録すれば、直ちに最適化計算を行い、その結果を確認することができます。

既存ブックの利用

最

最適化したいデータが含まれた Excel ブックがあれば、最適化モデルの各定数と Excel シート上のデータを簡易な操作で結びつけることができます。また、最適化モデルの各変数、目的関数についても、それらの値をシート上の任意の場所に表示することができます。

作業時間の短縮

入

入力データの登録、最適化計算の実行、結果の取込場所の指定などの作業に煩わされることはありません。データの作成、最適化モデルの構築、結果の分析などの本質的な作業に時間を費やすことができます。

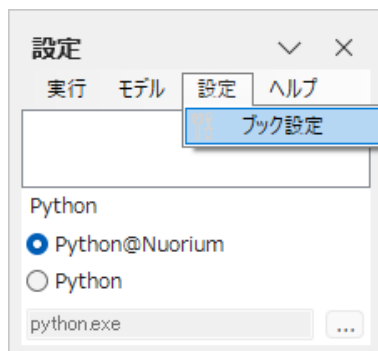
1.2 ご利用になる前に

Excel アドインを利用するには、まず Windows のログインユーザーごとに Excel アドインをインストールする必要があります。インストールの詳細については Nuorium Optimizer のインストールガイド^{*1}を参照してください。

◇ 1.2.1 Python 環境

PySIMPLE モデルと Excel を連携させるには Python 環境が必要です。Nuorium Optimizer 開発版には PySIMPLE がインストールされた Python 環境が同梱されているので、Python 環境の構築をせずに、そのままご利用いただけます。ただし、ご自身で構築した Python 環境を利用したい場合や Nuorium Optimizer ランタイム版をご利用の場合は、PySIMPLE がインストールされた Python 環境を構築する必要があります。

どの Python 環境を使うかはブックごとに設定する必要があります。リボンの「アドイン」→「Nuorium Optimizer」→「設定」で表示される設定画面のメニュー「設定」→「ブック設定」で、利用する Python 環境を選択することができます。



既定値は「Python@Nuorium」で Nuorium Optimizer に同梱された Python 環境を利用します。ご自身で構築した Python 環境を利用する場合や Nuorium Optimizer ランタイム版をご利用の場合は「Python」

^{*1} インストールガイドはインストールメディアまたはダウンロードサイト^{*2}にあります。

^{*2} <https://www.msi.co.jp/solution/nuopt/docs>

を選択して、その下にあるテキストボックスに利用する Python 環境の python.exe のパスを設定します (python.exe にパスが通っている場合は python.exe のみ)。

◇ 1.2.2 互換性について

Nuorium Optimizer V27 の Excel アドインより以下の機能が追加されました。

- PySIMPLE モデルに対応
- 複数モデルファイルの登録

V27 より前の Excel アドインで作成した Excel ブックは V27 以降の Excel アドインで開いて利用することができますが、一旦 V27 以降の Excel アドインで開いた Excel ブックは V27 より前の Excel アドインで開いて利用することはできません (設定が削除されます)。

2

チュートリアル

Excel アドインの基本的な使い方を説明します。**例題の紹介**と**モデルの記述**は最適化モデルの説明です。最適化モデルの説明が不要な方は**モデルを開く**から読み進めることができます。本チュートリアルで使うモデルファイル、Excel ブックはサンプルディレクトリ (Nuorium Optimizer のインストール先)^{*3} \Samples\ExcelAddIn にあります。"割当問題 (設定済み) .xlsx" は下記で説明している設定を終えた Excel ブックで、ブックを開いたら直ちに最適化を実行することができます。

2.1 例題の紹介

次の例題 (割当問題) を解いていくことにします。

^{*3} 初期設定では C:\Program Files (x86)\Mathematical Systems Inc\NUOPT です。

□ 例題（割当問題）

ある飲食店の従業員「安藤」「佐藤」「鈴木」「山本」「渡辺」の5人に仕事を割り当てます。仕事は「接客」「厨房」「レジ打ち」「仕入」「掃除」「仕込み」の6つです。各人を仕事に割り当てるにはコストがかかり、それは個人・仕事によって決まります。また、各人はそれぞれの仕事に対して熟練度があり、熟練度が高いほどコストがかかる傾向があります。以下は熟練度とコストをまとめたものです。

熟練度	安藤	佐藤	鈴木	山本	渡辺
接客	-1	3	-2	3	-4
厨房	5	-2	3	-4	5
レジ打ち	0	3	-2	3	-1
仕入	-3	-1	1	1	2
掃除	2	-2	2	-3	4
仕込み	5	-2	0	1	5

コスト	安藤	佐藤	鈴木	山本	渡辺
接客	570	1400	520	1410	450
厨房	1800	1000	1700	1050	2300
レジ打ち	800	1500	500	1500	600
仕入	500	600	1000	1000	1200
掃除	1200	500	1200	500	1300
仕込み	1500	1000	1200	1200	1500

また、割り当てる仕事を決定する際には以下の点を守らなくてはなりません。

- ・各人に割り振る仕事は最大で3つまでとする。
- ・「接客」「厨房」「レジ打ち」「掃除」「仕込み」は2人を割り当てる。
- ・「仕入」は1人を割り当てる。
- ・「接客」と「厨房」は別の人が担当する。
- ・各仕事について、担当する人の熟練度の和を、その仕事のクオリティとする。
- ・「接客」「厨房」のクオリティは2以上にする。
- ・「レジ打ち」「仕入」「掃除」「仕込み」のクオリティは0以上にする。

このとき、コストの合計を最小にするような割り当て方を求めてください。

2.2 モデルの記述

Nuorium Optimizer を用いるためには、解きたい問題を数理最適化問題として表現し、モデリング言語で記述する必要があります。ここでは **例題の紹介** で取り上げた例題を PySIMPLE で記述します。

本例題を数理最適化問題として表現すると次のようになります。

集合	
JOB = { 接客, 厨房, レジ打ち, 仕入, 掃除, 仕込み }	仕事の集合
PEOPLE = { 安藤, 佐藤, 鈴木, 山本, 渡辺 }	人の集合
変数 (0-1 整数変数)	
$x_{jp}, j \in \text{JOB}, p \in \text{PEOPLE}$	仕事 j を人 p に割り当てるならば $x_{jp} = 1$ 、そうでないならば $x_{jp} = 0$
定数	
$\text{cost}_{jp}, j \in \text{JOB}, p \in \text{PEOPLE}$	仕事 j を人 p に割り当てる際のコスト
$\text{jyukuren}_{jp}, j \in \text{JOB}, p \in \text{PEOPLE}$	仕事 j を人 p が行う際の熟練度
$\text{necessary}_j, j \in \text{JOB}$	仕事 j に割り振る必要がある人数
$\text{quality}_j, j \in \text{JOB}$	仕事 j に最低必要なクオリティ
目的関数 (最小化)	
$\sum_{j,p} \text{cost}_{jp} \times x_{jp}$	コストの総和
制約条件	
$\sum_p x_{jp} = \text{necessary}_j, \forall j \in \text{JOB}$	各仕事に必要な人数割り当てる
$\sum_j x_{jp} \leq 3, \forall p \in \text{PEOPLE}$	各人には、最大3つまでの仕事を割り当てることができる
$\sum_p \text{jyukuren}_{jp} \times x_{jp} \geq \text{quality}_j, \forall j \in \text{JOB}$	各仕事に必要なクオリティを確保する
$\sum_{j \in \{\text{接客, 厨房}\}} x_{jp} \leq 1, \forall p \in \text{PEOPLE}$	接客、厨房は違う人が担当する (同じ人が接客と厨房を兼ねない)

上記の数理最適化問題を PySIMPLE で記述すると、次のようになります。

```
1 from pysimple import *
2 from pysimple.ext import excel
3
4 # 問題の作成
5 problem = Problem()
6
7 # 集合の宣言
8 j = Element(dim=1, value=excel['仕事'])
9 p = Element(dim=1, value=excel['従業員'])
10
11 # 変数の宣言
12 excel['割り当て'] = x = BinaryVariable(index=(j,p))
13
14 # 定数の宣言
15 cost = Parameter(index=(j,p), value=excel['コスト'])
16 jyukuren = Parameter(index=(j,p), value=excel['熟練度'])
17 necessary = Parameter(index=j, value=excel['必要人数'])
18 quality = Parameter(index=j, value=excel['必要クオリティ'])
```

(次のページに続く)

(前のページからの続き)

```

19 # 各仕事に必要な人数割り当てる
20 problem += Sum(x[j,p], p) == necessary[j]
21 # 各人には、最大 3 つまでの仕事を割り当てることができる
22 problem += Sum(x[j,p], j) <= 3
23 # 各仕事に必要なクオリティを確保する
24 problem += Sum(x[j,p] * jyukuren[j,p], p) >= quality[j]
25 # 接客, 厨房は違う人が担当する
26 j2 = Element(value=['接客', '厨房'])
27 problem += Sum(x[j2,p], j2) <= 1
28
29 # 目的関数 (総コスト)
30 problem += Sum(cost[j,p]*x[j,p], (j,p)), "総コスト"
31
32 # 求解
33 problem.solve()
34
35 excel['総コスト'] = Parameter(value=problem.objective.val)
36

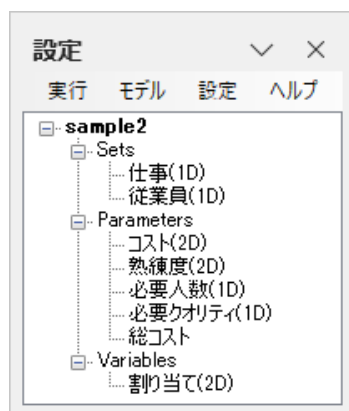
```

2.3 モデルを開く

Excel ブック "割当問題.xlsx" を開きます。リボンの「アドイン」→「Nuorium Optimizer」→「モデル」で表示されるファイル選択ダイアログボックスで sample2.py を選択します。



最適化モデルが開かれると定数一覧、変数一覧といったモデル情報が読み込まれ、ブックの左側に下図のような設定画面が表示されます。なお、設定画面のメニューはリボンと同等の内容です。



2.4 データの登録

「仕事」、「従業員」、「コスト」、「熟練度」、「必要人数」、「必要クオリティ」に対して Excel シートの範囲を登録します。

熟練度	安藤	佐藤	鈴木	山本	渡辺
接客	-1	3	-2	3	-4
厨房	5	-2	3	-4	5
レジ打ち	0	3	-2	3	-1
仕入	-3	-1	1	1	2
掃除	2	-2	2	-3	4
仕込み	5	-2	0	1	5
コスト	安藤	佐藤	鈴木	山本	渡辺
接客	570	1400	520	1410	450
厨房	1800	1000	1700	1050	2300
レジ打ち	800	1500	500	1500	600
仕入	500	600	1000	1000	1200
掃除	1200	500	1200	500	1300
仕込み	1500	1000	1200	1200	1500
	必要人数	必要クオリティ			
接客	2	2			
厨房	2	2			
レジ打ち	2	0			
仕入	1	0			
掃除	2	0			
仕込み	2	0			

まずは「仕事」の範囲を選択した状態で設定画面のモデルツリーの「仕事 (1D)」の右クリックメニュー → 「選択範囲を入力」をクリックします。

行 モデル 設定 ヘルプ

sample2

- Sets
 - 仕事(1D)
 - 従業員(1D)
 - 選択範囲を入力
 - Parameters
 - コスト(2D)
 - 熟練度(2D)

4		
5		接客
6		厨房
7		レジ打ち
8		仕入
9		掃除
10		仕込み

「従業員」についても同様に登録します。

sample2

- Sets
 - 仕事(1D)
 - 入力
 - 従業員(1D)
 - 選択範囲を入力
 - Parameters
 - コスト(2D)

4		熟練度	安藤	佐藤	鈴木	山本	渡辺
5		接客	-1	3	-2	3	-4
6		厨房	5	-2	3	-4	5
7		レジ打ち	0	3	-2	3	-1
8		仕入	-3	-1	1	1	2

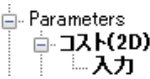
次に「コスト」の表全体を選択した状態で設定画面のモデルツリーの「コスト (2D)」の右クリックメニュー → 「選択範囲を入力」をクリックします。

sample2

- Sets
 - 仕事(1D)
 - 入力
 - 従業員(1D)
 - 入力
 - Parameters
 - コスト(2D)
 - 選択範囲を入力
 - Variables
 - 必要人数(1D)
 - 必要クオリティ(1D)
 - 総コスト

12		コスト	安藤	佐藤	鈴木	山本	渡辺
13		接客	570	1400	520	1410	450
14		厨房	1800	1000	1700	1050	2300
15		レジ打ち	800	1500	500	1500	600
16		仕入	500	600	1000	1000	1200
17		掃除	1200	500	1200	500	1300
18		仕込み	1500	1000	1200	1200	1500

「コスト」に入力データとして選択範囲が登録され、モデルツリーの「コスト (2D)」は下図のようになります。

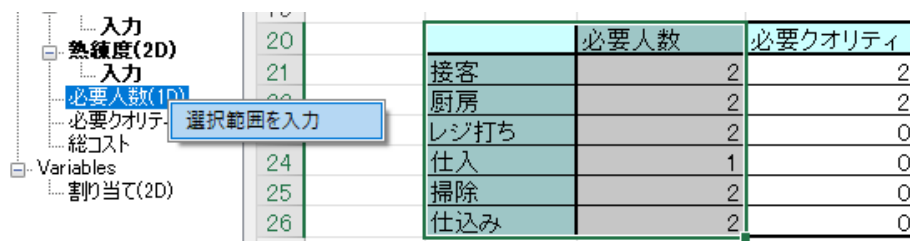


「熟練度」、「必要人数」、「必要クオリティ」についても同様に登録します。なお、「必要クオリティ」については添字と値の範囲が離れていますが、添字と値全体を含む形で範囲を選択します。

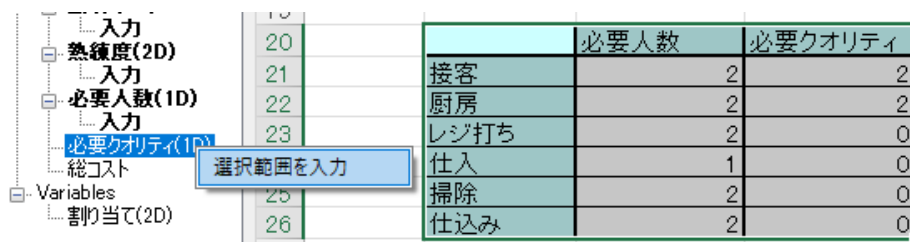
sample2

- Sets
 - 仕事(1D)
 - 入力
 - 従業員(1D)
 - 入力
 - Parameters
 - コスト(2D)
 - 入力
 - 必要人数(1D)
 - 選択範囲を入力

4		熟練度	安藤	佐藤	鈴木	山本	渡辺
5		接客	-1	3	-2	3	-4
6		厨房	5	-2	3	-4	5
7		レジ打ち	0	3	-2	3	-1
8		仕入	-3	-1	1	1	2
9		掃除	2	-2	2	-3	4
10		仕込み	5	-2	0	1	5
11							



	必要人数	必要クオリティ
接客	2	2
厨房	2	2
レジ打ち	2	0
仕入	1	0
掃除	2	0
仕込み	2	0



	必要人数	必要クオリティ
接客	2	2
厨房	2	2
レジ打ち	2	0
仕入	1	0
掃除	2	0
仕込み	2	0

上記の全ての登録を終えるとモデルツリーは下図のようになります。



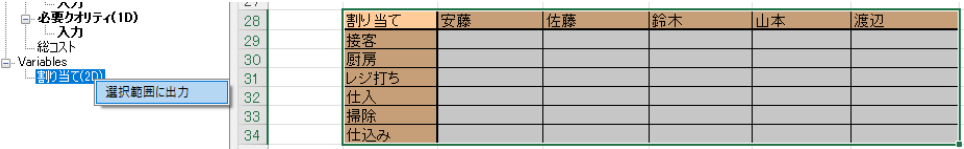
それぞれの名前の下にある「入力」ノードをクリックすると登録されている範囲を確認することができます。

2.6 結果の出力

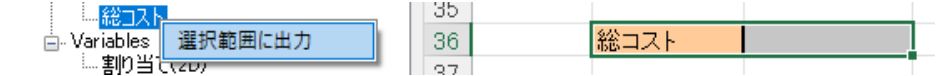
「割り当て」、「総コスト」に対して結果を出力するよう Excel シートの範囲を登録します。

割り当て	安藤	佐藤	鈴木	山本	渡辺
接客					
厨房					
レジ打ち					
仕入					
掃除					
仕込み					
総コスト					

まずは「割り当て」の表全体を選択した状態で設定画面のモデルツリーの「割り当て」の右クリックメニュー → 「選択範囲に出力」をクリックします。



「総コスト」についても同様に登録します。



上記の「割り当て」、「総コスト」が登録されるとモデルツリーは下図のようになります。



次回の実行からは上で登録した範囲に結果が出力されます。

割り当て	安藤	佐藤	鈴木	山本	渡辺
接客	1	0	0	1	0
厨房	0	1	0	0	1
レジ打ち	0	1	1	0	0
仕入	0	0	0	1	0
掃除	1	1	0	0	0
仕込み	0	0	1	1	0
総コスト	12380				

「割り当て」、「総コスト」の下にある「出力」ノードをクリックすると登録されている範囲を確認することができます。

最適化モデル

Nuorium Optimizer を使って最適化をするには最適化モデルが必要です。Nuorium Optimizer V27 の Excel アドインから PySIMPLE で書いた最適化モデルと連携できるようになりました。また、1つの Excel ブックに複数の最適化モデルが登録できるようになりました。

3.1 モデルの書き方

Excel と連携させるための最適化モデルの書き方を説明します。

◇ 3.1.1 PySIMPLE

まずは、PySIMPLE モデルを Excel と連携させるために `pysimple.ext` モジュールの `excel` オブジェクトを `import` します。

```
from pysimple.ext import excel
```

この `import` した `excel` オブジェクトを介して Excel とデータのやり取りをします。

次に、Excel にあるデータを `Parameter` に入力したい場合は、`Parameter` の定義時に以下のように書きます。

```
cost = Parameter(index=(j,p), value=excel['コスト'])
```

Excel のモデルツリーに「コスト」ノードが表示され Excel 上のデータを登録できるようになります。

また、最適化結果の `Variable` の値を Excel に出力する場合は、`Variable` の定義時に以下のように書きます。

```
excel['割り当て'] = x = BinaryVariable(index=(j,p))
```

Excel のモデルツリーに「割り当て」ノードが表示され Excel 上に出力する場所を登録できるようになります。

なお、Excel と連携する **Parameter** や **Variable** の **index** に指定されている **Element** には **dim** を明示的に指定する必要があります。

```
j = Element(dim=1, value=excel['仕事'])
p = Element(dim=1, value=excel['従業員'])
```

Element の **set** に指定されている **Set** についても同様です。

◇ 3.1.2 C++SIMPLE

C++SIMPLE モデルについては基本的に Excel との連携のための変更は必要ありません。モデルに書いた **Parameter**、**Variable**、**Objective** など Excel と連携可能なオブジェクトは Excel のモデルツリーに全て表示されます。

3.2 モデルを開く

最初に最適化モデルファイル（PySIMPLE の場合は .py ファイル、C++SIMPLE の場合は .smp ファイルまたは mknuopt.bat で作成した .exe ファイル）を開き^{*4}、モデル情報を取得する必要があります。リボンの「アドイン」→「Nuorium Optimizer」→「モデル」で表示されるファイル選択ダイアログボックスでモデルファイルを選択します。



モデル情報が読み込まれると、設定画面にモデルツリーが表示されます。

^{*4} C++SIMPLE の資源制約付きスケジューリング問題を記述したモデル（rcpsp モデル）は開けません。



モデルファイルはブックに紐付きます。1つのブックに複数のモデルファイルを紐付けることができます。モデルノードの右クリックメニューの「詳細」で設定画面下部にモデルの詳細画面が表示されます。

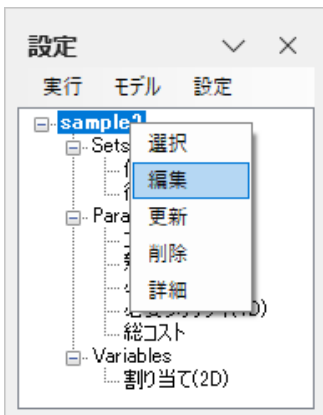


モデルの詳細画面では、モデルの表示名、モデルファイルのパス、相対パスか否かを設定します。モデルファイルが Excel ブックと同じディレクトリまたはそのサブディレクトリに存在する場合は、モデルファイルのパスは相対パスとして扱われます。その他の場合は絶対パスとして扱われますが「相対」をチェックするとブックからの相対パスとして扱うことになります。

Excel ブックとモデルファイルを一緒に動かすような場合は「相対」をチェックする、モデルファイルを常に同じ場所に置いておくような場合は「相対」のチェックをはずす、というように使い分けることができます。

3.3 モデルの編集

選択されているモデルが .py ファイル (PySIMPLE) または .smp ファイル (C++SIMPLE) の場合はモデルノードの右クリックメニューの「編集」で Nuorium によりモデルを編集することができます。



3.4 モデルの更新

モデルノードの右クリックメニューの「更新」で編集したモデルのモデル情報を更新することができます。モデルを編集して保存してもモデル情報は自動では更新されない点に注意してください。



4

データの登録

最適化モデルの入力データについて説明します。

4.1 範囲を登録

Excel シート上のデータを最適化モデルの入力データとして登録することができます。基本的な操作方を以下に示します。シート上の範囲を選択します。

コスト	安藤	佐藤	鈴木	山本	渡辺
接客	570	1400	520	1410	450
厨房	1800	1000	1700	1050	2300
レジ打ち	800	1500	500	1500	600
仕入	500	600	1000	1000	1200
掃除	1200	500	1200	500	1300
仕込み	1500	1000	1200	1200	1500

モデルツリー上の SIMPLE オブジェクトの右クリックメニューで「選択範囲を入力」をクリックします。

sample2

Sets

仕事(ID)

従業員(ID)

Parameters

熱伝度(1)

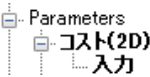
必要人数(ID)

必要クオリティ(ID)

選択範囲を入力

12	コスト	安藤	佐藤	鈴木	山本	渡辺
13	接客	570	1400	520	1410	450
14	厨房	1800	1000	1700	1050	2300
15	レジ打ち	800	1500	500	1500	600
16	仕入	500	600	1000	1000	1200
17	掃除	1200	500	1200	500	1300
18	仕込み	1500	1000	1200	1200	1500

選択範囲が SIMPLE オブジェクトの入力データとして登録され、モデルツリーに「入力」ノードが追加されます。



登録と同時に添字範囲と値範囲が解釈され、それぞれ選択されて表示されます。

添字1

添字2

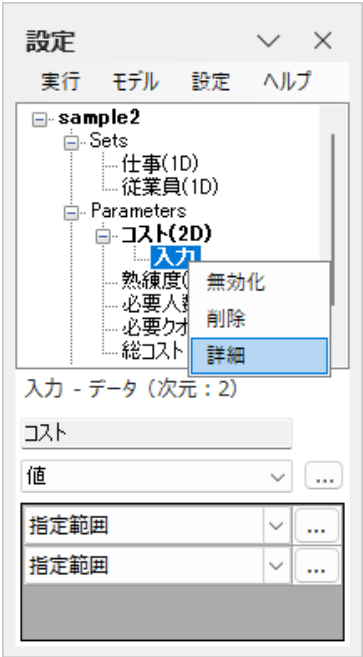
コスト	安藤	佐藤	鈴木	山本	渡辺
接客	570	1400	520	1410	450
厨房	1800	1000	1700	1050	2300
レジ打ち	800	1500	500	1500	600
仕入	500	600	1000	1000	1200
掃除	1200	500	1200	500	1300
仕込み	1500	1000	1200	1200	1500

値

1つのSIMPLEオブジェクトに対して複数の「入力」ノードを登録することができますが、有効なノードは1つだけです。有効なノードは太字で表示されます。「入力」ノードの右クリックメニューの「有効化」、「無効化」でそのノードの有効／無効を選択することができます。

4.2 範囲の詳細設定

登録された範囲は詳細を確認したり、修正することができます。追加された「入力」ノードをダブルクリックまたは右クリックメニュー → 「詳細」をクリックするとモデルツリーの下に詳細設定画面が現れます。



値範囲や添字範囲をクリックし、それぞれ登録されている範囲を確認することができます。

入力 - データ (次元: 2)	12	コスト	安藤	佐藤	鈴木	山本	渡辺
コスト	13	接客	570	1400	520	1410	450
値	14	厨房	1800	1000	1700	1050	2300
指定範囲	15	レジ打ち	800	1500	500	1500	600
指定範囲	16	仕入	500	600	1000	1000	1200
	17	掃除	1200	500	1200	500	1300
	18	仕込み	1500	1000	1200	1200	1500

入力 - データ (次元: 2)	12	コスト	安藤	佐藤	鈴木	山本	渡辺
コスト	13	接客	570	1400	520	1410	450
値	14	厨房	1800	1000	1700	1050	2300
指定範囲	15	レジ打ち	800	1500	500	1500	600
指定範囲	16	仕入	500	600	1000	1000	1200
	17	掃除	1200	500	1200	500	1300
	18	仕込み	1500	1000	1200	1200	1500

入力 - データ (次元: 2)	12	コスト	安藤	佐藤	鈴木	山本	渡辺
コスト	13	接客	570	1400	520	1410	450
値	14	厨房	1800	1000	1700	1050	2300
指定範囲	15	レジ打ち	800	1500	500	1500	600
指定範囲	16	仕入	500	600	1000	1000	1200
	17	掃除	1200	500	1200	500	1300
	18	仕込み	1500	1000	1200	1200	1500

編集ボタンを押すことにより登録範囲を修正することができます。

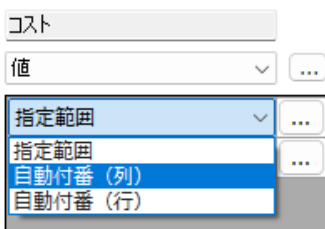
入力 - データ (次元 : 2)



4.3 自動付番

添字範囲は「自動付番 (列)」または「自動付番 (行)」に変更することができます。

入力 - データ (次元 : 2)



添字範囲として「自動付番 (列)」が選択された場合は 1,2,... (値範囲の行数)、「自動付番 (行)」が選択された場合は 1,2,... (値範囲の列数) が添字の中身として指定されたことになります。

入力 - データ (次元 : 2)

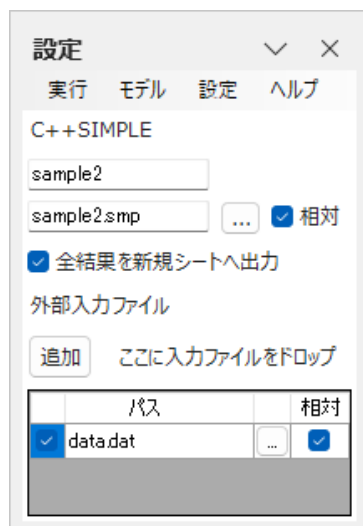
コスト					
値					
指定範囲					
自動付番 (列)					
自動付番 (行)					

		1	2	3	4	5
40	1	570	1400	520	1410	450
41	2	1800	1000	1700	1050	2300
42	3	800	1500	500	1500	600
43	4	500	600	1000	1000	1200
44	5	1200	500	1200	500	1300
45	6	1500	1000	1200	1200	1500
46						

4.4 外部入力ファイル

本機能は C++SIMPLE モデルでのみ有効です。

Excel シート上で登録するデータの他に外部のデータファイル (.dat ファイルまたは .csv ファイル) を最適化モデルの入力データとして登録することができます。モデルノードの右クリックメニューの「詳細」で詳細画面が表示されます。「外部入力ファイル」の「追加」ボタンを押すかデータファイルをドラッグ&ドロップすることにより外部の入力ファイルを追加することができます。



モデルファイルパスと同様に「相対」列にチェックを入れることによりデータファイルのパスは相対パスとして扱われます。

5

実行

最適化の実行について説明します。

5.1 実行

リボンの「アドイン」→「Nuorium Optimizer」→「実行」で最適化計算が始まります。



最適化計算実行中は下図のような実行画面が表示されます。

実行

sample2

バージョン

Nuorium Optimizer ...

ステータス

最適化正常終了

問題名

Problem

変数の数

30

制約式の数

22

整数変数の数

30

目的

最小化

アルゴリズム

単体法+分枝限定法

問題種別

線形整数計画

目的関数値

12380

部分問題数

5

実行可能解の個数

3

メモリ利用状況(Mbyte)

499/0(134212882/0)

経過時間(秒)

0.07

<iteration end>

<<wls end>>

#2

12480

-inf

+inf

0.1

0

499

sol: wls

<<hsimplex begin>>

dual-phase2 start

Iter.

Objective

Primal Inf.

Dual Inf.

Time(s)

0

1.052308e+00

1.056642e+01

0.000000e+00

0.1

16

1.051520e+01

0.000000e+00

0.000000e+00

0.1

cleanup perturbation

16

1.051516e+01

0.000000e+00

0.000000e+00

0.1

<<hsimplex end>>

#3

12380

11400

7.916

0.1

1

499

sol: local

12380

12380

0.000

0.1

0

499

<iteration end>

[Result]

STATUS

OPTIMAL

VALUE_OF_OBJECTIVE

12380

SIMPLEX_PIVOT_COUNT

79

PARTIAL_PROBLEM_COUNT

5

ELAPSED_TIME(sec.)

0.07

writing: C:\Users*opt-div\Desktop\ExcelAddin\sample2_pdata\result\result.json

実行画面には以下の機能があります。

モデル選択コンボボックス

実

行するモデルを選択します。

実行ボタン

モ

デルの実行、実行中は停止、強制終了を行います。

取込ボタン

実

行結果を取り込みます。

クリアボタン

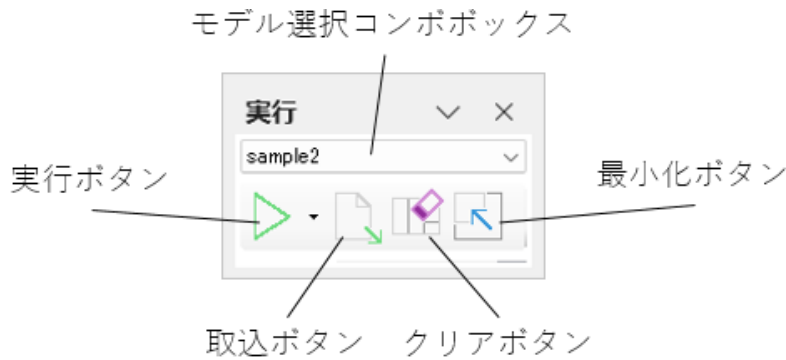
取

り込んだ実行結果をクリアします。

最小化ボタン

実

行画面を最小化します。



5.2 停止

実行画面の「停止」ボタンを押すと最適化計算を途中で終了させることができます。このとき最適化計算のアルゴリズムによっては途中の結果を得ることができます。

実行

markshare2

☐ 停止
☒ 強制終了

Nuorium Optimizer ...
分枝限定法による整数...

問題名	markshare2
変数の数	60
制約式の数	7
整数変数の数	60
目的	最小化
アルゴリズム	単体法+分枝限定法
問題種別	線形整数計画
下界値	3.637978807e-12
上界値	20
部分問題数	1056765
活性部分問題の数	4215
実行可能解の個数	7
メモリ利用状況(Mbyte)	44/0(134213420/0)
上界, 下界ギャップ	20
経過時間(秒)	39.69

```
# (Hard/Soft) Penalty      = 0 / 0
# Elapsed Time(s)          = 0.968 / 0.982
# Iterations                = 9813 / 10000
=====
<iteration end>
<<wls end>>
#3          49          -inf          +inf          1.0          0          30 sol: wls

<<hsimplex begin>>
dual-phase2 start
  Iter.   Objective   Primal Inf.   Dual Inf.   Time(s)
    0   -6.000013e+01   2.901807e+02   0.000000e+00   1.0
   18   -2.939437e+01   0.000000e+00   0.000000e+00   1.0
cleanup perturbation
   18   -2.939429e+01   0.000000e+00   0.000000e+00   1.0
<<hsimplex end>>

          49   1.81899e-12   100.000          1.0          1          30 cut: 12
          49   3.63798e-12   100.000          1.7          1          30 cut: 6
          39   3.63798e-12   100.000          1.8          3          30 sol: rins
          34   3.63798e-12   100.000          3.8         1427          34 sol: relax
          32   3.63798e-12   100.000         13.8         2035          37 sol: relax
          20   3.63798e-12   100.000         17.7         3461          40 sol: relax
          20   3.63798e-12   100.000         32.7         3807          42
```

最適化計算のアルゴリズム及びタイミングによっては上記の「停止」ボタンで最適化計算を終了できない場合があります。実行画面の「停止」→「強制終了」ボタンを押すと最適化計算を強制的に終了させることができます。この場合は途中の結果を得ることはできません。

6

結果の出力

最適化の結果出力について説明します。

6.1 新規シートに出力

本機能は C++SIMPLE モデルでのみ有効です。

モデルの詳細画面の「全結果を新規シートへ出力」にチェックを入れると最適化の結果は新規シートに書かれます。

C++SIMPLE

sample2(C)

sample2.smp ... ☒ 相対

☒ 全結果を新規シートへ出力

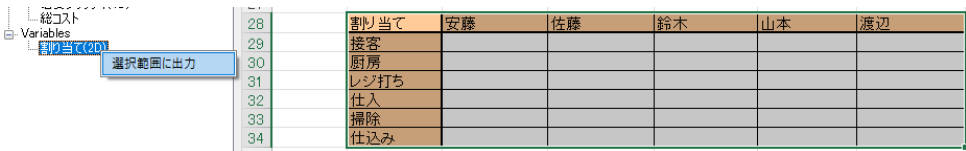
初期設定では上記のチェックは入った状態です。結果を Excel シート上の範囲に出力する設定を行うと上記のチェックは自動的に外れます。

6.2 範囲に出力

最適化の結果を Excel シート上の範囲に出力できます。基本的な操作方法を以下に示します。シート上の範囲を選択します。

割り当て	安藤	佐藤	鈴木	山本	渡辺
接客					
厨房					
レジ打ち					
仕入					
掃除					
仕込み					

モデルツリー上の SIMPLE オブジェクトの右クリックメニューで「選択範囲に出力」をクリックします。



選択範囲が SIMPLE オブジェクトの出力として登録され、モデルツリーに「出力」ノードが追加されます。



登録と同時に添字範囲と値範囲が解釈され、それぞれ選択されて表示されます。

添字1 添字2

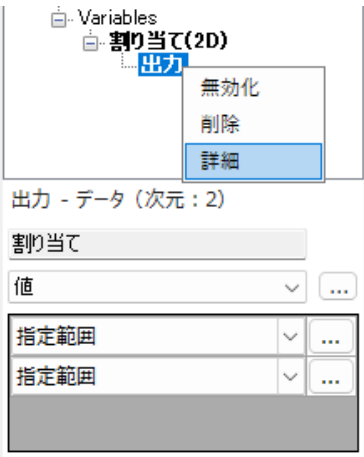
割り当て	安藤	佐藤	鈴木	山本	渡辺
接客					
厨房					
レジ打ち					
仕入					
掃除					
仕込み					

値

1つの SIMPLE オブジェクトに対して複数の「出力」ノードを登録することができます。有効なノードは太字で表示されます。「出力」ノードの右クリックメニューの「有効化」、「無効化」でそのノードの有効／無効を選択することができます。

6.3 範囲の詳細設定

登録された範囲は詳細を確認したり、修正することができます。追加された「出力」ノードをダブルクリックまたは右クリックメニュー → 「詳細」をクリックするとモデルツリーの下に詳細画面が現れます。



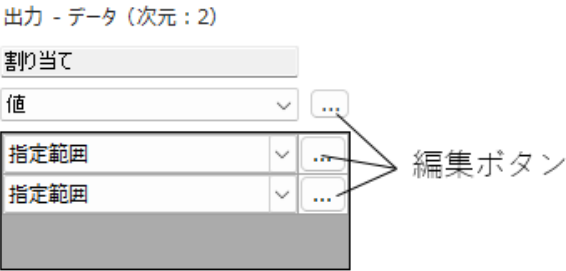
値範囲や添字範囲をクリックし、それぞれ登録されている範囲を確認することができます。

出力 - データ (次元: 2)	28	割り当て	安藤	佐藤	鈴木	山本	渡辺
割り当て	29	接客					
値	30	厨房					
指定範囲	31	レジ打ち					
指定範囲	32	仕入					
	33	掃除					
	34	仕込み					

出力 - データ (次元: 2)	28	割り当て	安藤	佐藤	鈴木	山本	渡辺
割り当て	29	接客					
値	30	厨房					
指定範囲	31	レジ打ち					
指定範囲	32	仕入					
	33	掃除					
	34	仕込み					

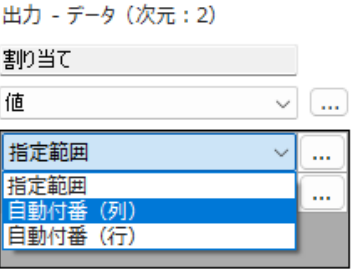
出力 - データ (次元: 2)	28	割り当て	安藤	佐藤	鈴木	山本	渡辺
割り当て	29	接客					
値	30	厨房					
指定範囲	31	レジ打ち					
指定範囲	32	仕入					
	33	掃除					
	34	仕込み					

編集ボタンを押すことにより登録範囲を修正することができます。



6.4 自動付番

入力範囲と同様に添字範囲は「自動付番 (列)」または「自動付番 (行)」に変更することができます。



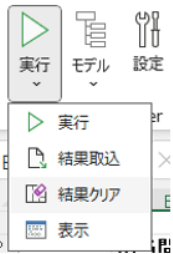
添字範囲として「自動付番 (列)」が選択された場合は1,2,... (値範囲の行数)、「自動付番 (行)」が選択された場合は1,2,... (値範囲の列数) が添字の中身として指定されたことになります。

実行	モデル	設定	ヘルプ	39	1	2	3	4	5
出力 - データ (次元 : 2)				40	1				
割り当て				41	2				
値				42	3				
指定範囲				43	4				
自動付番 (列)				44	5				
自動付番 (行)				45	6				
				46					
				47					

6.5 結果のクリア

リボンの「アドイン」→「Nuorium Optimizer」→「実行」→「結果クリア」で出力設定した範囲の値をクリアすることができます。

割り当て	安藤	佐藤	鈴木	山本	渡辺
接客	1	0	0	1	0
厨房	0	1	0	0	1
レジ打ち	0	1	1	0	0
仕入	0	0	0	1	0
掃除	1	1	0	0	0
仕込み	0	0	1	1	0
総コスト	12380				

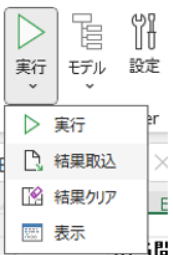


割り当て	安藤	佐藤	鈴木	山本	渡辺
接客					
厨房					
レジ打ち					
仕入					
掃除					
仕込み					
総コスト					

6.6 結果の取込

最適化計算を実行すると次の実行まで結果はファイルとして保存されています。リボンの「アドイン」→「Nuorium Optimizer」→「実行」→「結果取込」で保存されている結果を出力設定した範囲に読み込むことができます。

割り当て	安藤	佐藤	鈴木	山本	渡辺
接客					
厨房					
レジ打ち					
仕入					
掃除					
仕込み					
総コスト					



割り当て	安藤	佐藤	鈴木	山本	渡辺
接客	1	0	0	1	0
厨房	0	1	0	0	1
レジ打ち	0	1	1	0	0
仕入	0	0	0	1	0
掃除	1	1	0	0	0
仕込み	0	0	1	1	0
総コスト	12380				

6.7 solve インデックス

本機能は C++SIMPLE モデルでのみ有効です。
最適化モデルの中で複数回 solve したとき、結果に solve 番号が現れます。

solve 番号 添字 値

solve	j	x
1	銘柄A	0.074666
1	銘柄B	0.198642
1	銘柄C	0.060309
1	銘柄D	0.666383
2	銘柄A	0.064018
2	銘柄B	0.113427
2	銘柄C	0.380517
2	銘柄D	0.442038

「solve インデックスを追加」をチェックすることにより、solve 番号をインデックスとして追加することができます。変数 x は 1 次元（添字が 1 つ）のデータですが solve インデックスを追加することにより、2 次元のデータとして扱われていることに注意してください。

出力 - データ（次元：1）

x

値

☒ Solve インデックス

指定範囲

指定範囲

x	銘柄A	銘柄B	銘柄C	銘柄D
1	0.074666	0.198642	0.060309	0.666383
2	0.064018	0.113427	0.380517	0.442038

6.8 下限値／上限値／双対値

オブジェクトの種類や求解アルゴリズムによっては、結果の値のほかに下限値／上限値／双対値が得られることがあります。

	値	下限値	上限値	双対値
j	x	lower	upper	dual
銘柄A	0.074666	0	1.#INF	0
銘柄B	0.198642	0	1.#INF	0
銘柄C	0.060309	0	1.#INF	0
銘柄D	0.666383	0	1.#INF	0

これら結果の種類を指定してシート上の範囲に表示することができます。

出力 - データ (次元: 1)

x

値

j	値	下限値	上限値	双対値
銘柄A	0.064018	0	1.#INF	0
銘柄B	0.113427	0	1.#INF	0
銘柄C	0.380517	0	1.#INF	0
銘柄D	0.442038	0	1.#INF	0

出力 - データ (次元: 1)

x

下限値

j	値	下限値	上限値	双対値
銘柄A	0.064018	0	1.#INF	0
銘柄B	0.113427	0	1.#INF	0
銘柄C	0.380517	0	1.#INF	0
銘柄D	0.442038	0	1.#INF	0

出力 - データ (次元: 1)

x

上限値

j	値	下限値	上限値	双対値
銘柄A	0.064018	0	1.#INF	0
銘柄B	0.113427	0	1.#INF	0
銘柄C	0.380517	0	1.#INF	0
銘柄D	0.442038	0	1.#INF	0

出力 - データ (次元: 1)

x

双対値

j	値	下限値	上限値	双対値
銘柄A	0.064018	0	1.#INF	0
銘柄B	0.113427	0	1.#INF	0
銘柄C	0.380517	0	1.#INF	0
銘柄D	0.442038	0	1.#INF	0