

航空宇宙実習 事前トレーニング

演習マニュアル

航空宇宙のPCが必要です

anaconda
python 3.9
Jupyter

2023年9月6日版

作成

原 武史（岐阜大学）

前提条件

1. ファイルの保存場所（パス）を理解している
2. OneDriveなどのオンラインドライブを利用していないドライブに保存できる
3. 環境構築が終わっている
4. Jupyter Notebookの使い方がわかる
5. 高精度な結果を得るのは簡単じゃないよ～ を理解している！

内 容

- Lesson 0 事前準備：データのダウンロード
- Lesson 1 線形計画法
- Lesson 2 ナップザック問題
- Lesson 3 割当問題
- Lesson 4 Bin Packing／Nesting
- Lesson 5 量子計算

事前準備

Step 1 ファイルをダウンロードする

Teamsもしくは以下からダウンロードする.

<http://micv.sakura.ne.jp/prg/archives/308>

ダウンロード先は個別に指定します.

ファイルは4つ.

CHECK	CHECK.zip	チェックスクリプト
LPNS	LPNS.zip	線形計画法関連
BIN	Bin.zip	Bin Packing関連
QC	QC.zip	量子コンピューティング関連

Step 2 ファイルを展開して保存する

ローカルのドライブに必ず保存する

ZIPファイルを確実に解凍する

ZIPファイルのまま中身が見れる状態では何もできません

Step 3 中身を確認する

ファイルを保存した場所（例えばデスクトップ）を見る

そこにフォルダを作って保存. 場所をよく覚えておく.

Step 4 ターミナルで次のコマンドを投入

```
pip install pulp
pip install ortoolpy
pip install dwave-ocean-sdk
```

Step 5 チェックスクリプトを動かす

Jupyter Notebookを起動する

保存したディレクトリに移動して, チェックスクリプトを動かす

Shift+Returnで動くよ

importの3行がエラーを出さなければOK

Step 5 完璧！（のはず）

環境構築の際のチェックスクリプトが動くはず. 動けば, 完璧なはず！！

Lesson1：線形計画法

複数の素材から最適な生産数を導出する：LP.ipynb

課題1：調理

4種類の素材：素材1，素材2，素材3，素材4がある。
3種類の料理：料理1，料理2，料理3がある。
すべての料理は，4種類の素材の配合を変えて作る。
それぞれの料理の値段は，価格1，価格2，価格3（料理の数）である。
総売上を最大にするには，料理をどれだけ作ったらよいか？

（条件1：準備された素材の量の単位）

素材1の量：20
素材2の量：40
素材3の量：50
素材4の量：80

（条件2：製法の単位）

料理1 素材1：素材2：素材3 = 3：1：2：4
料理2 素材1：素材2：素材3 = 1：5：3：1
料理3 素材1：素材2：素材3 = 3：3：2：2

（条件3：値段）

料理1 500円
料理2 800円
料理3 400円

（問題）

1. pythonのプログラムを参考に実行手順を理解しましょう
2. 得られた結果は何を意味するか？
3. 素材の配合を変えて結果を調べましょう
4. 素材の配合や価格を変えて最適化結果を調べましょう
5. 応用事例を調べましょう

Lesson2：ナップサック問題

入れる重さに制限があるナップサックの中にもっとも価値を詰め込む：

NS.ipynb

items.csv

課題1：つめこみ

(条件1：荷物の条件)

以下の荷物がある。価値と重さがわかっている。それぞれ1つずつある。

荷物	価値	重さ
----	----	----

1	110	12
---	-----	----

2	130	14
---	-----	----

3	80	7
---	----	---

4	100	9
---	-----	---

5	250	22
---	-----	----

6	185	16
---	-----	----

7	150	15
---	-----	----

(条件2：ナップサックの制限)

ナップサックには重さ70までしか入れられない。

(条件3：荷物の大きさ)

荷物の大きさは考えなくてよい。

(問題)

1. どの荷物を入れるとよいか？
2. その時の荷物の重さの合計は？
3. ナップサックの制限が変わったら組み合わせが変わるか？
4. 価値が高く、かつ、軽い荷物が増えたらどうなるか？
5. 価値が低く、かつ重い荷物はいつ選ばれるか？
6. 荷物の条件を自由に変えて実行してみよう

Lesson3：割当問題

適材適所を自動的に

LOC.ipynb

Score.csv

課題1：適材適所

(条件)

4種類の部品を4か所に配置する.

それぞれの部品をそれぞれに配置したときの適性を表にスコアで示す

適性は高い方がよい

適性が0もある.

各場所には必ず1つ配置する.

各部品は必ず1か所に収まる.

	部品1	部品2	部品3	部品4
場所1	5	4	0	3
場所2	5	4	4	3
場所3	0	5	4	5
場所4	2	0	0	3

条件のもと、適性の和が最大になるような配置を定めなさい.

(問題)

1. どのような配置が定まるか？
2. スコアを変えたらどうなるか？
3. 場所と部品を自由に増やしなさい

Lesson4 : Bin Packing

NestingをBin Packingで実現 : BIN.zipを解凍してください

プログラム : binpacking.ipynb

課題1 : どれだけ詰め込めるか

ナップサック問題と似ているがこちらの方が単純.

(条件)

ビンと同じ大きさ／長さ

できるだけ少ないビンにいろいろな大きさの部品を詰め込みたい

詰め込むことは, 切り出すことと考えられる

切り出すときには, 刃幅を部品に入れておく必要がある

部品の数よりも切り出す回数は1回多い (一回は両端を切る)

ビンの大きさからその一回分は引いておいてもよい

(問題)

次の長さの部品を切り出したい.

```
ws = [521, 511, 501, 501, 491, 491, 491, 481, 481, 481, \
      471, 471, 471, 471, 471, 471, 471, 461, 461, 461, \
      461, 451, 451, 451, 451, 441, 421, 431, 431, 431, \
      230, 311]
```

ただし, 1本の部材 (ビン) の長さは3650である.

できるだけ少ない部材で実現したい.

どのように切り出したらよいか?

条件のもと, 切り出す配置を定めなさい.

Lesson5 : Traveling Saleman Problem: TSP

TSPを量子コンピューティングでやってみよう

QC.zipを解凍してください.

プログラム : Dwave01.ipynb

課題1 : どんなルートがよいのか？

(条件)

都市間の距離はすべてわかっている.

距離は行列で保存されている.

できるだけ少ない距離ですべての都市を1回訪問したい.

都市間の距離は次の通りである.

同じ都市間の距離は大きな値 (1000) として表現した.

	A	B	C	D	E
A	1000	20	20	50	40
B	30	1000	10	30	20
C	20	10	1000	30	20
D	50	30	20	1000	10
E	40	20	20	10	1000

(問題)

5つの都市を巡回したい. どのような順に訪問すればよいか？

補足：量子コンピューティング

以下を参考に、量子コンピューティングでBin packingをやってみよう.

https://docs.ocean.dwavesys.com/en/stable/examples/hybrid_cqm_binpacking.html

D-Waveのその他のサンプルにもチャレンジしよう.