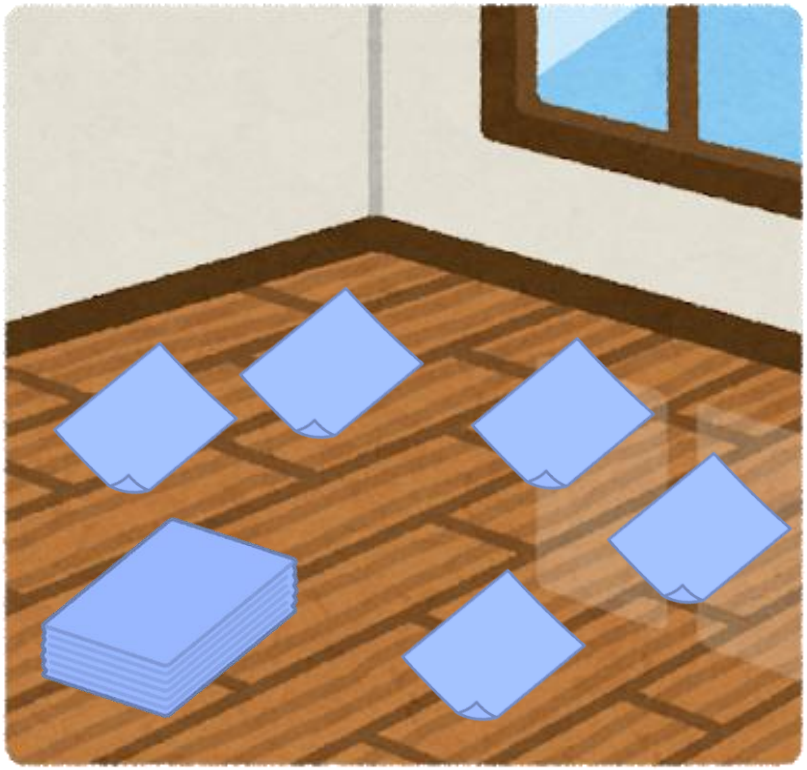
2021年3月31日

床に紙を並べる

otosula

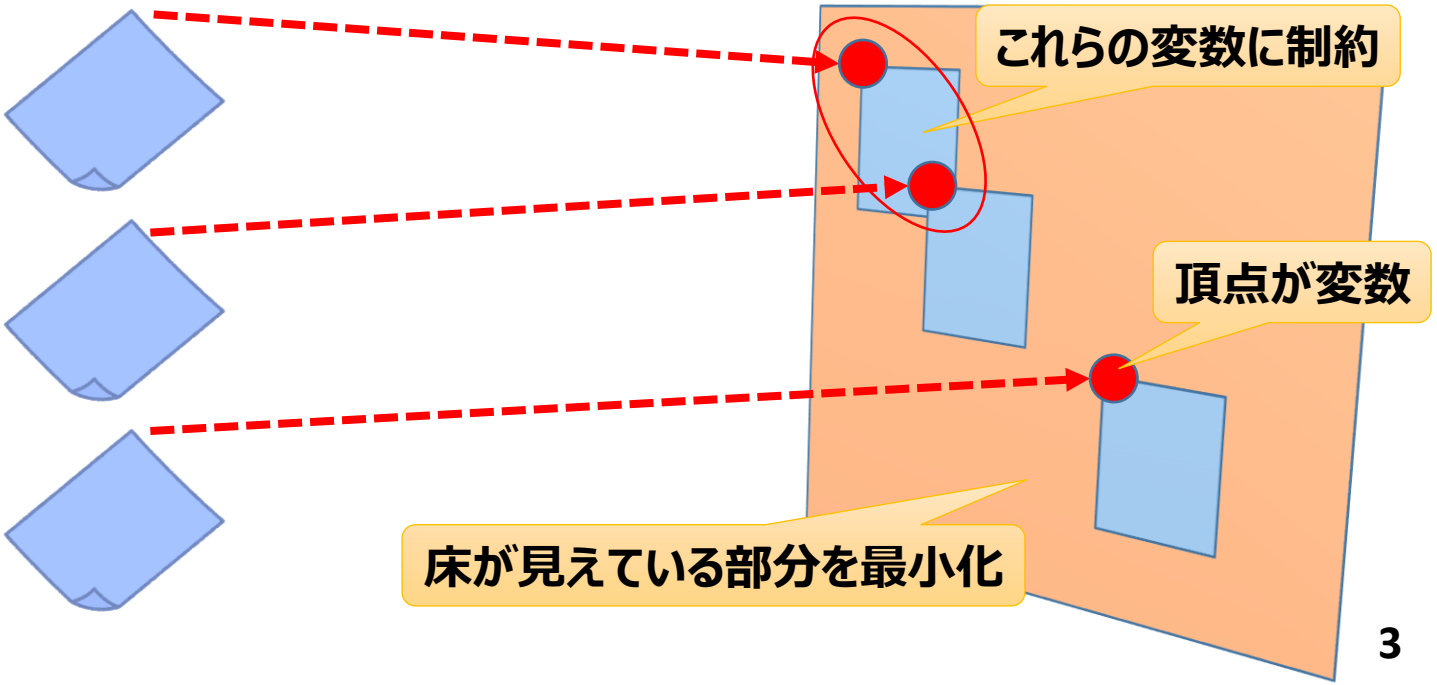
解決する問題

・床をなるべく覆い隠すように紙を置きたい。
・但し、几帳面で節約家なので、紙は重ならないようにしたい。



基本のアイデア

- 紙の頂点の位置を決めると、床の占有範囲が決まる。
- 床の占有箇所を共有する紙の頂点の集合にone-hot等の制約を掛ければ、紙を重ならずに置ける。
- また、目的関数は見えている床の範囲を最小化すればよい。



変数定義

- 床
  - 四角形。縦と横の長さで定義する。
    floor = [9, 16]

- 紙
  - こちらも四角形。紙の種類はいくつか用意する。
    paper = [[3, 4], [2, 3], [5, 1]]

  - 紙は回転して使えるので、縦と横を入れ替えた要素を追加する。
    paper = [[3, 4], [2, 3], [5, 1], [4, 3], [3, 2], [1, 5]]

  - 各紙の枚数を決めておく（問題なく床を埋められるくらいの数）
    paper_num = 3

  - 各紙は床上の頂点の位置を持つ（決定変数）
    q_paper = gen_symbols(BinaryPoly, (len(paper) * paper_num, floor[0], floor[1]))

制約条件：紙を重ねない制約

- 全ての紙について、床の頂点の位置をズラしながら、床の占有範囲を求め、床の各箇所において、重なる紙の頂点を集める。

```
for i in range(len(paper) * paper_num):
    for p_v, p_h in itertools.product(range(floor[0]), range(floor[1])):
        for f_v, f_h in itertools.product(range(p_v, p_v + p_size_v), range(p_h, p_h + p_size_h)):
            q_floor[f_v][f_h] += q_paper[i][p_v][p_h]
```

※ p_size_vとp_size_hは対象の紙のサイズ

※ q_floor[f_v][f_h]は床の各箇所の頂点を集める変数

全ての紙に

床の位置をズラしながら

床の占有範囲を求め

重なる紙の頂点を求める

- 床の各箇所において、重なる紙の頂点にone-hot制約を掛けようかと思ったけど、床に紙を置かないケースが考えられるのでless_equal制約を掛ける。

```
for f_v, f_h in itertools.product(range(floor[0]), range(floor[1])):
    g_floor = less_equal(q_floor[f_v][f_h], 1)
```

制約条件：紙を使えるのは一度という制約

- 各紙の、床の頂点を決める全ての変数に、one-hot制約を掛けようかと思ったけど、床に紙を置かないケースが考えられるのでless_equal制約を掛ける。

```
for i in range(len(paper) * paper_num):
    for p_v, p_h in itertools.product(range(floor[0]), range(floor[1])):
        q_paper += q_paper[i][p_v][p_h]
    g_use += less_equal(g_paper, 1)
```

各紙の

床の頂点を決める全ての変数に

less_equal制約を掛ける

目的関数

- 全ての紙について、その紙が使われた場合（=その紙の頂点を決める変数の合計が0でなければ）、その紙の面積が求まり、それを足し合わせる。

```
for i in range(len(paper) * paper_num):
    for p_v, p_h in itertools.product(range(floor[0]), range(floor[1])):
        p_area += q_paper[i][p_v][p_h]
    p_area * (p_size_v * p_size_h)
    all_p_area += p_area
```

全ての紙について

その紙が使われた場合

その紙の面積が求まり、

それを足し合わせる

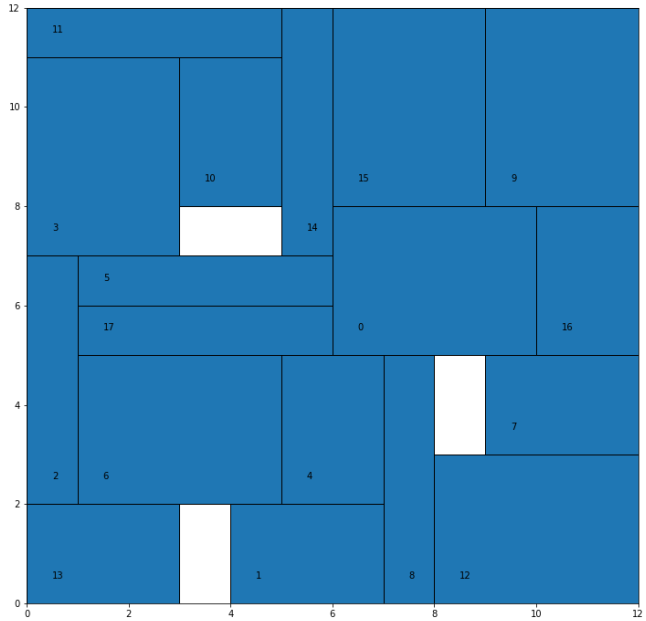
- 床の面積から足し合わせた紙の面積を引いた値を最小化する

```
objective = ((floor[0] * floor[1]) - all_p_area) ** 2
```

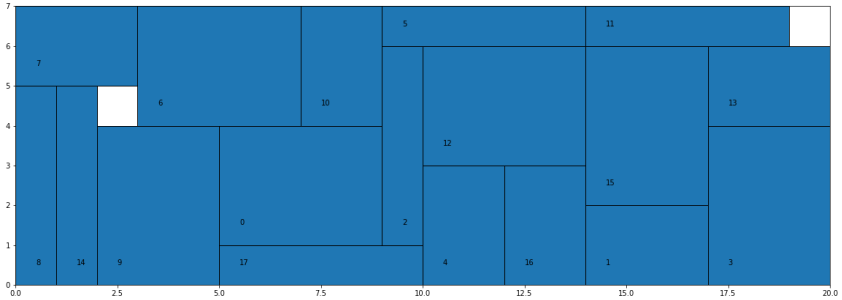
結果

paper = [[3, 4], [2, 3], [5, 1]]
paper_num = 3

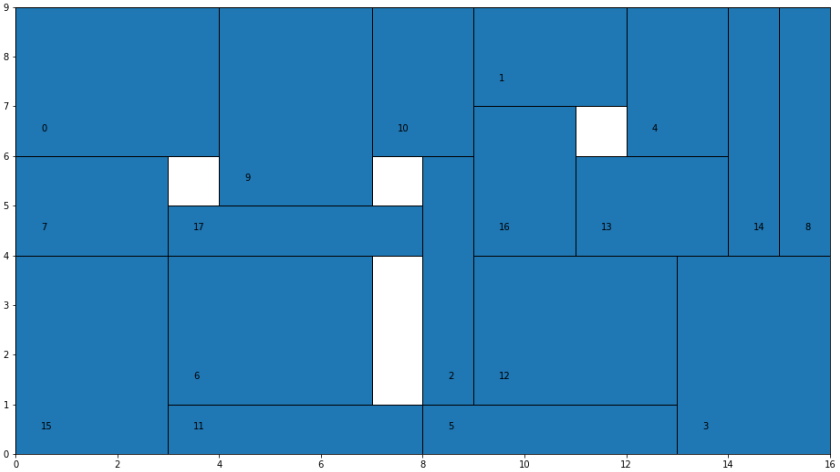
floor = [12, 12]



floor = [7, 20]



floor = [9, 16]



その他

- たまに解が出ない。
- 実は制約には重み（x10）を付けているが、付けるべきなのか、付けるならどれくらいの値なのか、全く理解していない。
- 本当は部屋を箱で埋めたかった。