\Orchestrating a brighter world

本棚に本を見栄え良く配置する

NECソリューションイノベータ株式会社
橋本　慎司　・　柴田　将
高原　玲央　・　森　英憲
今井　春奈

\Orchestrating a brighter world

NECは、安全・安心・公平・効率という
社会価値を創造し、
誰もが人間性を十分に発揮できる
持続可能な社会の実現を目指します。

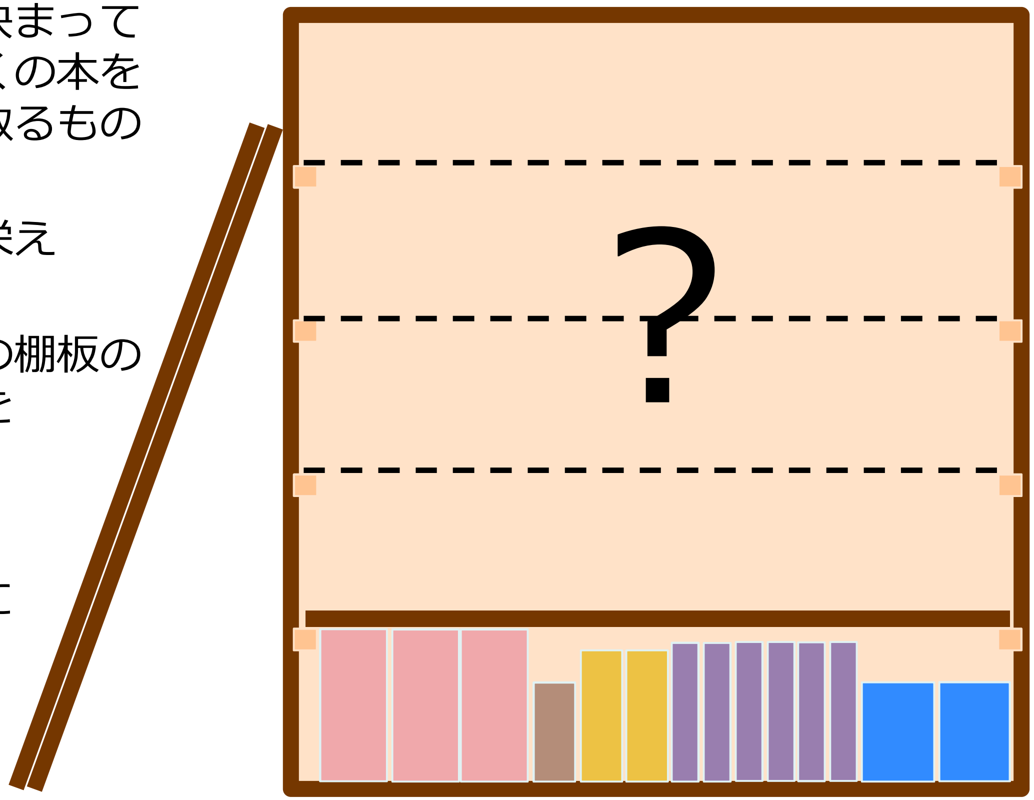
取り上げた問題

概要

- 設置可能な棚板の位置と棚板の数が決まっている本棚があり、その本棚により多くの本を収納したい。ただし、本はよく手に取るものをなるべく収納したい。
- 多くの本を収納しつつ、なるべく見栄えが良くなるように収納したい。
- 多くの本を見栄え良く収納するための棚板の配置と本の配置の最適な組み合わせをAmplifyを用いて解いてみる。

前提条件

- 本を横に倒し、本と棚板のスペースに収納することは考えない。
- 棚板自体の太さ、耐えられる重さは考慮しない。



設定データ

データ

- 本棚
  - サイズ（横幅、高さ）
  - 棚板の数と設置できる位置

- 本
  - タイトル
  - シリーズ
  - ジャンル（新書、単行本など）
  - 著者名
  - サイズ（厚さ、高さ）
  - 使用する頻度

目的関数の考え方

- 本ごとに使用頻度を設けて、使用頻度に見栄えの要素を追加した総合点数で良し悪しを判断する。

- 見栄えは、本の属性がなるべく棚ごとに揃っているほど良いものとする。

- 重視する見栄えは以下とする。

1. シリーズ 高
2. ジャンル
3. 高さ
4. 著者名（今回は使用せず） 低

解法

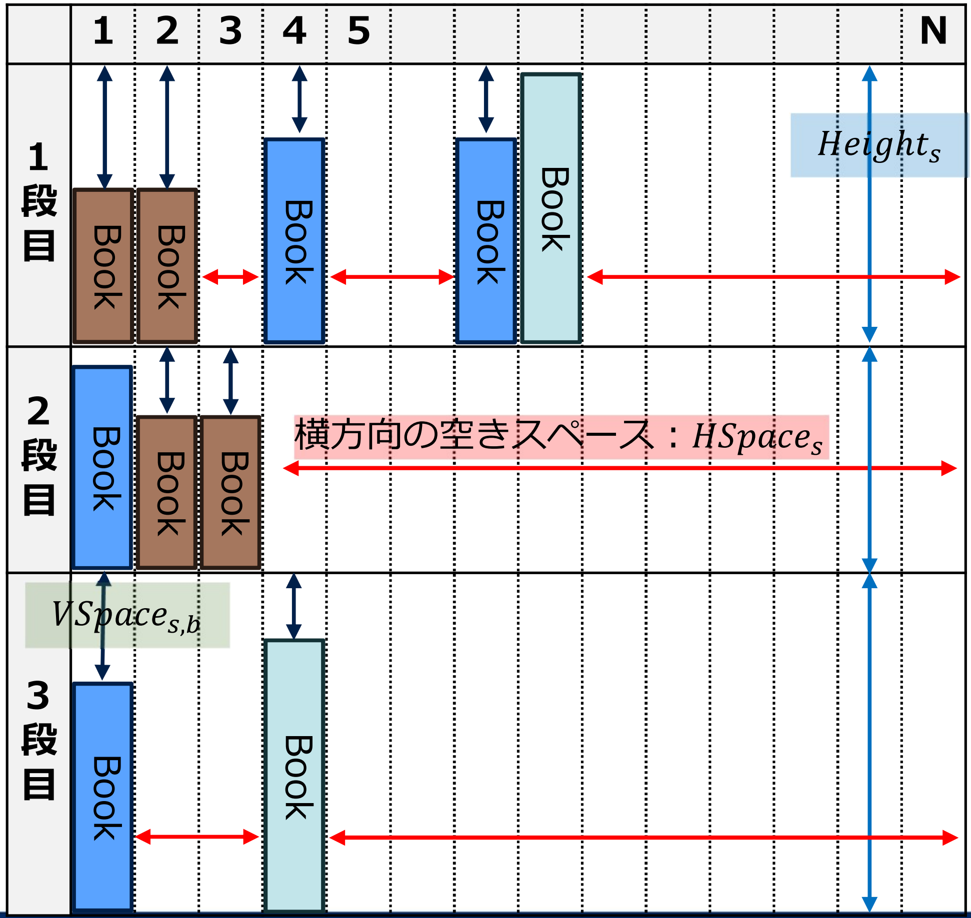
考えかた

- １回のアニーリングで本の配置と棚板の設置位置をまとめて決定する。
- 各棚板に本を収納する「枠」を用意して、その枠に本を当てはめるか否かの組み合わせとして解く。

変数の定義

- $x_{b,s,m}$ ：本（*b*）は、上から*s*段目の左から*m*番目に配置するか否かを表す変数
  ※*m*は、対象となる本の厚さから、一番多く本を配置できる場合の数とする。
- $y_{s,n}$ ：本を配置する際に、各段の高さの合計が高さを超えないようにするための格段の高さを表す変数
- $S1_{s,n}$ ：本を配置する際に、横幅を超えないようにするスラック変数
- $S2_{s,n}$ ：各段に格納される本が、各段の高さを超えないようにするスラック変数

変数の考え方



変数定義

$x_{b,s,i}$ ： 本 b を s 段目 の i 番目 に配置
$y_{s,i}$ ： 各段の高さを表す
$S1_{s,i}$ ： 各段の空きスペースを表す
$S2_{s,i}$ ： 各本と棚板の間の空きスペースを表す

$$Height_s = \sum_i 2^i \, y_{s,i}$$

$$HSpace_s = \sum_i 2^i \, S1_{s,i}$$

$$VSpace_{s,b} = \sum_{i \in I_b} 2^i \, S2_{s,i}$$

定式化

目的関数

- 本棚に詰める本を最適化

$$Obj1 = \sum_{b}\sum_{s}\sum_{m} freq(b) \times x_{b,s,m}$$

- ただし、見栄えも考慮する

$$Obj2 = \sum_{b1,b2}\sum_{s}\sum_{m} x_{b1,s,m} \times x_{b2,s,m+1}$$ （$b1, b2$は同じジャンルや同じシリーズ）

$$Obj3 = \sum_{s}\sum_{m}\left(\sum_{b} x_{b,s,m} \times BookHeight(b) - \sum_{b} x_{b,s,m+1} \times BookHeight(b)\right)^2$$

目的関数＝$-(Obj1 + Obj2 + Obj3)$

定式化

制約条件

- 条件式１：本はどこか１か所に配置されるか、どこにも配置されない。

$$H_1 = \sum_{b \in Books} \left\{ \left(\sum_{s \in Shelves} \sum_{n} x_{b,s,m} - 0.5 \right)^2 - 0.25 \right\}$$

- 条件式２：ある位置には本は１冊配置されるか、何も配置されない。

$$H_2 = \sum_{s \in Shelves} \sum_{n} \left\{ \left(\sum_{b \in Books} x_{b,s,m} - 0.5 \right)^2 - 0.25 \right\}$$

- 条件式３：棚の幅以上の本が収納されない。

$$H_3 = \sum_{s \in Shalves} \left(ShelfWidth - \sum_{m} x_{b,s,m} \times BookWidth(b) - S1_{s,n} \right)^2$$

定式化

制約条件

- 条件式４：各段の高さの合計は棚の高さに一致する。

$$H_4 = \left(ShelfHeight \quad - \sum_{s} y_{s,n} \right)^2$$

- 条件式５：各段に格納される本は、それぞれの段の高さを超えない。

$$H_5 = \sum_{b \in Books} \sum_{s \in Shelves} \left(\sum_{m} x_{b,s,m} \times \left(\sum_{s} y_{s,n} - BookHeight_b - S2_{s,n} \right) \right)^2$$

課題

- 定式化したところ、変数の数が非常に大きくなることにより適切な解をうまく求めることができない問題が発生。
次頁以降の解法１，２により簡略化して解を求めることとした。

解法 1

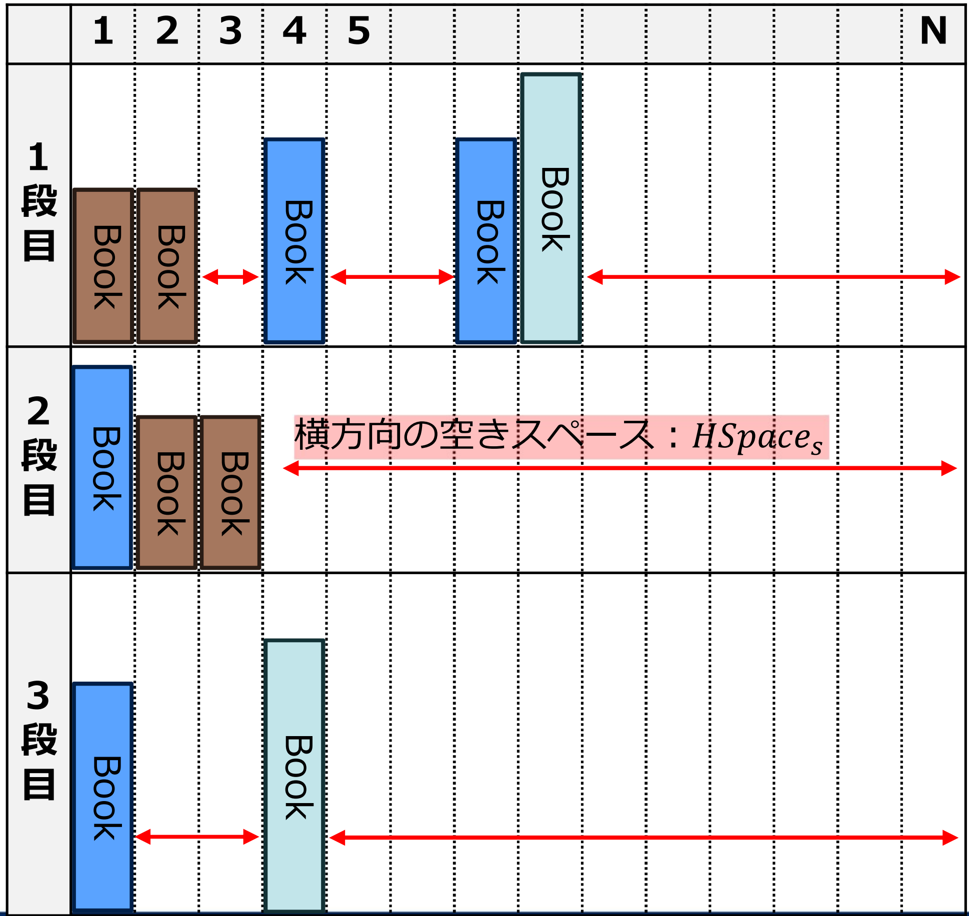
考えかた

- 変数$x_{b,s,m}$の考え方は変更しない。ただし、以下の方針により変数の数を削減する。
  - シリーズ本は同じ（隣り合う）位置に配置するのが最優先であるため、シリーズ本は1冊の大きな本としてとらえて配置を考える。この対応により、本の数が削減可能。（かつ副産物としてシリーズ本に関するソフト制約式も不要になる）
  - 棚の位置とサイズを固定して解くこととし、棚の全組み合わせでコストを比較することとする。この対応により、棚の位置が決定するため、各段の高さの組み合わせを求める必要がなくなり、スピンを削減可能。

変数の定義

- $x_{b,s,m}$：本（*b*）は、上から*s*段目の左から*m*番目に配置するか否かを表す変数
  ※*m*は、対象となる本の厚さから、一番多く本を配置できる場合の数とする。
- $S1_{s,n}$：本を配置する際に、横幅を超えないようにするスラック変数

解法１の変数の考え方



変数定義

$x_{b,s,i}$： 本 b を s 段目 の i 番目 に配置

$y_{s,i}$ ： 各段の高さを表す

$S1_{s,i}$： 各段の空きスペースを表す

$S2_{s,i}$： 各本と棚板の間の空きスペースを表す

$$Height_s = \sum_i 2^i \, y_{s,i}$$

$$HSpace_s = \sum_i 2^i \, S1_{s,i}$$

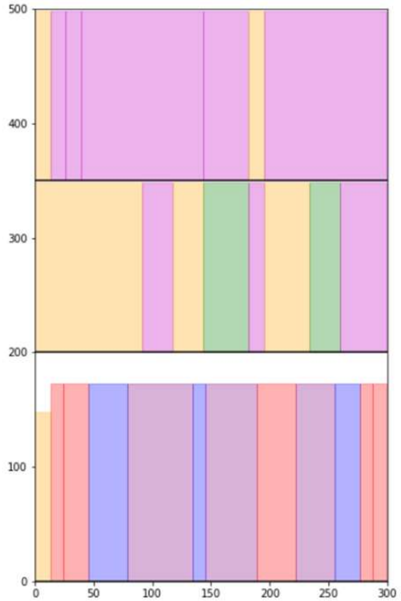
$$VSpace_{s,b} = \sum_{i \in I_b} 2^i \, S2_{s,i}$$

高さの概念を無くす

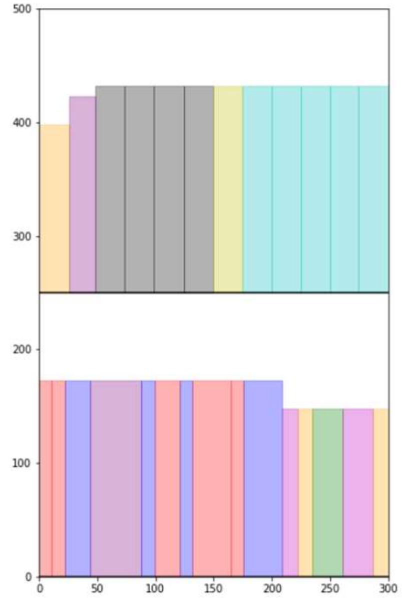
解法１の結果

使用した棚データ

- 棚の高さ：500mm
- 棚の幅：300mm
- 棚の配置パターン（棚板の高さを固定）
  - パターン１：150mm/150mm/200mm
  - パターン２：250mm/250mm
- 収納対象の本の数：200



パターン１



パターン２

今回用意したデータを解法１で求解した場合は、パターン１の棚の配置パターンでより適した本の配置を求めることができた。

但し、棚の配置を固定しており、結果は必ずしも良いものとは言えず。

解法２

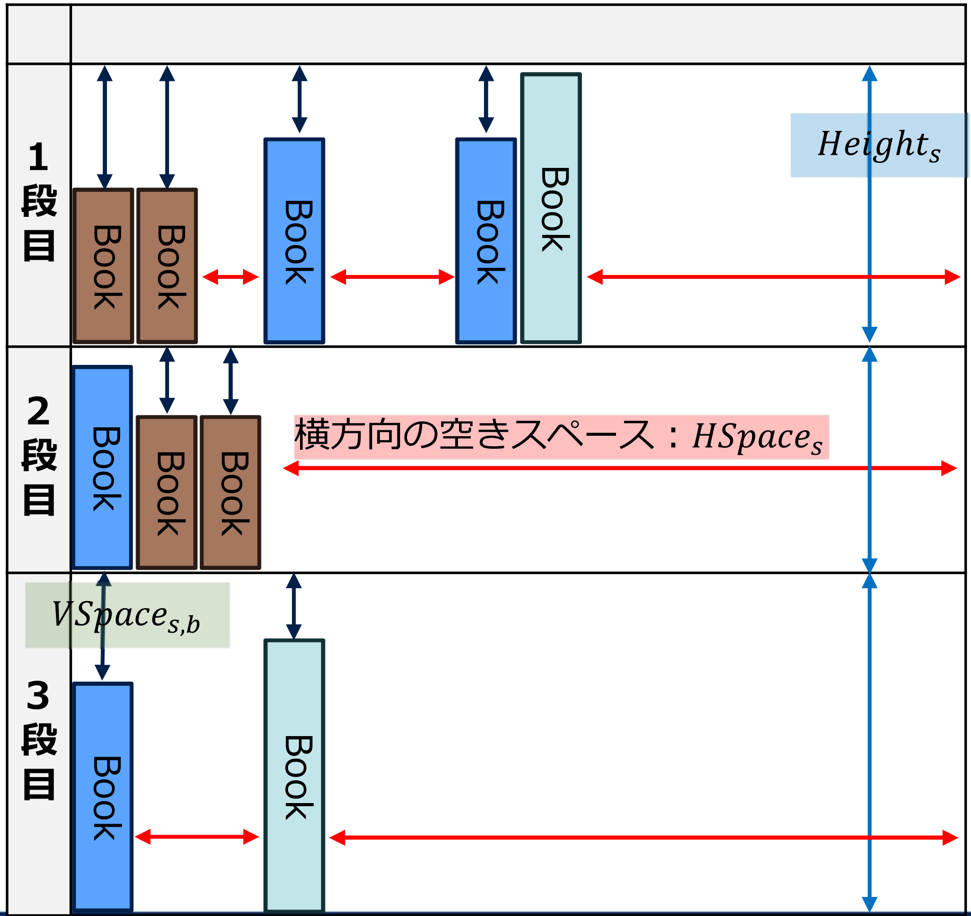
考え方

- 問題を2段階で解く。
- まず、本棚の高さと幅に収まるように本を振り分ける部分をAmplifyで求解する。
- 求解した後、各棚板に振り分けた本を、単純なソート処理によって、ペナルティが少なくなるように配置位置を決定する。

変数の定義

- $x_{b,s}$：上からs段目の棚に本（b）を収納する。
- $y_{s,n}$：本を配置する際に、各段の高さの合計が高さを超えないようにするための格段の高さを表す変数
- $S1_{s,n}$：本を配置する際に、横幅を超えないようにするスラック変数
- $S2_{s,n}$：各段に格納される本が、各段の高さを超えないようにするスラック変数

解法２の変数の考え方



変数定義

$x_{b,s,i}$： 本 b を s 段目 の i 番目 に配置
$y_{s,i}$ ： 各段の高さを表す
$S1_{s,i}$： 各段の空きスペースを表す
$S2_{s,i}$： 各本と棚板の間の空きスペースを表す

$$Height_s = \sum_i 2^i\, y_{s,i}$$

$$HSpace_s = \sum_i 2^i\, S1_{s,i}$$

$$VSpace_{s,b} = \sum_{i \in I_b} 2^i\, S2_{s,i}$$

枠（配置順）の概念を無くす
（ナップザックに似た考え方）

解法 2 の定式化

目的関数

- 使用頻度の大きい本を優先的に配置する

$$Obj1 = \sum_{b} \sum_{s} freq(b) \times x_{b,s}$$

- 同じジャンルの本は同じ段に入れやすくする

$$Obj2 = \sum_{b1,b2} \sum_{s} x_{b1,s} \times x_{b2,s} \quad (b1, b2\text{は同じジャンル})$$

$$\text{目的関数} = -(Obj1 + Obj2)$$

解法２の定式化

- 条件式１：本はどこかの棚に配置されるか、どこにも配置されない。

$$H_1 = \sum_{b \in Books} \left\{ \left(\sum_{s \in Shelves} \sum_{n} x_{b,s} - 0.5 \right)^2 - 0.25 \right\}$$

- 条件式２：棚の幅以上の本が収納されない。

$$H_2 = \sum_{s \in Shalves} \left(ShelfWidth - \sum_{b} \left(x_{s,b} \times BookWidth_b \right) - \sum_{i} S1_{s,i} \times 2^i \right)^2$$

解法２の定式化

- 条件式３：各段の高さの合計は棚の高さに一致する。

$$H_3 = \left(ShelfHeight - \sum_s \sum_i y_{s,i} \times 2^i \right)^2$$

- 条件式４：各段に格納される本は、それぞれの段の高さを超えない。

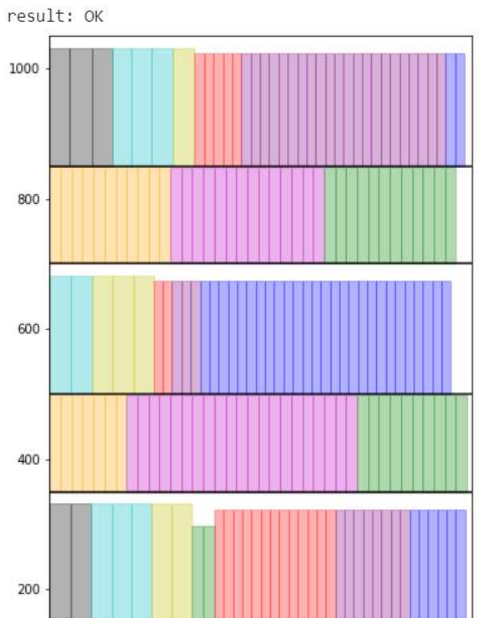
$$H_4 = \sum_{b \in Books} \sum_{s \in Shalves} x_{s,b} \times \left(\sum_i y_{s,i} \times 2^i - BookHeight(b) - \sum_i S2_{s,i} \times 2^i \right)^2$$

- 目的関数と条件式１～４で各段に本を割り当てた後、ジャンル順、タイトル順にソートした結果を最終解とする。

解法２の結果

検証に使用した棚データ

- 棚の高さ：1050mm
- 棚の幅：500mm
- 棚の数：6
- 棚の高さの設定単位：50mm刻み
- 収納対象の本の数：500



一部単純ソートにより並べ替えを行っているものの、解法１に比べて棚の配置位置も求解できた。今回は対応できなかったが、割り当てた後も量子アニーリングにより求解できれば、見栄えの制約がより複雑になった場合にも、柔軟に対応ができるのではと考える。

おわり

課題・所感

- 時間内に当初予定していたものを検証を終えることができず残念。
  - 解法1、解法２ともに志半ばとなった。
  - 目的関数として著者名を含めることもできなかった。
  - 検証に使用したデータも少なく、汎用性も疑問。様々なデータを使って求解できるかは不透明。
- 解法２では、スラック変数が多く、制約式のバランス（重み）の調整が難しかった。
- 本問題は、規模を拡大して図書館などで利用者が利用しやすいような配置を考えたり、本だけに限らず、食器棚など格納するサイズが変動するものへの最適な配置を求解するシーンに応用が利くのかも？

\Orchestrating a brighter world

NEC