# Shellの使い方

まずSCIPのWindowsバイナリをダウンロードすると、shellしかない。というわけで、とりあえずshellを使ってみよう。このフォルダにある**「scip-3.1.0.win.x86\_64.msvc.opt.spx.mt.exe」**をダブルクリックして起動すると、shellが開く。

## MPSファイル

基本的に、事前にファイルにinteger programmingの定義を記述し、それを***read***コマンドで読んで***optimize***し、***display solution***で結果を表示するという流れになる。対応しているファイルフォーマットの１つがMPS（Mathematical Programming System）だ。Wikipediaにも説明があるが、多少の方言があるかも知れない。以下のinteger programmingに対応するMPSファイル例を使って説明する。

NAME KNAPSACK

OBJSENSE

MAX ← Object関数をmaximizeしたい（デフォルトはminimize）

ROWS

N OBJ ← NはObjective関数を意味し、その後に、関数名が続く。

L LIM1 ← Lは制約関数がであることを意味し、その後に、制約関数名が続く。

制約関数が等式の場合は「E」、の場合は「G」を指定する。

COLUMNS ← 各制約式をこの中に記述する

MARK0000 'MARKER' 'INTORG' ← 以降の変数はintegerだということ。

X1 OBJ -4 LIM1 2 ← 2列に分けて考えてね。1列目には、objective関数の係数が、

X2 OBJ -5 LIM1 3 2列目には、制約関数の係数が、記述されている。

X3 OBJ -12 LIM1 5 1列目と2列目の意味的な違いはない。

X4 OBJ -14 LIM1 6

MARK0001 'MARKER' 'INTEND' ← ここまでの変数はintegerだということ。

RHS ← 制約関数の右辺を記述する。

RHS LIM1 9

BOUNDS

UP bnd X1 1 ← 各変数の範囲（1以下ということ）

UP bnd X2 1

UP bnd X3 1

UP bnd X4 1

ENDATA

MPSファイルの用意ができたら、後はshellで読み込んでみよう。

SCIP> **read knapsack.mps**

...

SCIP> **optimize**

...

SCIP> **display solution**

Objective value: 19

X2 1 (obj:5)

X4 1 (obj:14)

確かに、optimal solutionが得られた。objective関数は19で、、その他の変数は0という結果だ。

## Linear Programming

なお、上のMPSファイルで、MARK…の行を削除すれば、そのままlinear programmingの定義ファイルとして使える。SCIPのshellも、そのまま同じコマンドでlinear programmingを解いてくれる。

SCIP> **read knapsack\_lp.mps**

...

SCIP> **optimize**

...

SCIP> **display solution**

Objective value: 21.3333333

X2 1 (obj:12)

X4 0.66666667 (obj:14)

当然ながら、結果は異なる。

どのぐらい性能さがあるのかはまだ分からない。もっと大きい問題でやってみないと差がない。

## People Allocation

とりあえず使えることが分かったので、people allocation問題を解いてみよう。

まず、あるゾーンプランにおいて、各セルでのユーザ３人によるスコアは以下の通りとなった。各セル内で、１行目はユーザ１、２行目はユーザ２、３行目はユーザ３によるスコアだ。

|  |  |  |  |
| --- | --- | --- | --- |
| 0.15577334  0.24872956  0.23486331 |  |  |  |
| 0.36423028  0.18830030  0.31007376 | 0.35494941  0.45251310  0.23486331 |  |  |
|  | 0.36423028  0.18830030  0.31007376 | 0.35494941  0.45251310  0.23486331 | -0.00044122338  -0.18480703  -0.072083220 |
| 0.30757457  0.091772422  0.32012302 | 0.32162699  0.24430312  0.31949946 |  | 0.24086335  0.16167119  0.23486331 |

Integer programmingの形式で表すと、

対応するMPSファイルは、以下のようになる。

NAME PEOPLE\_ALLOCATION

OBJSENSE

MAX

ROWS

N OBJ

E LIM1

E LIM2

E LIM3

E LIM4

E LIM5

E LIM6

E LIM7

E LIM8

E LIM9

E LIM10

E LIM11

E LIM12

COLUMNS

MARK0000 'MARKER' 'INTORG'

X11 OBJ 0.15577334 LIM1 1

X11 LIM4 1

X12 OBJ 0.36423028 LIM1 1

X12 LIM5 1

X13 OBJ 0.35494941 LIM1 1

X13 LIM6 1

X14 OBJ 0.36423028 LIM1 1

X14 LIM7 1

X15 OBJ 0.35494941 LIM1 1

X15 LIM8 1

X16 OBJ -0.00044122338 LIM1 1

X16 LIM9 1

X17 OBJ 0.30757457 LIM1 1

X17 LIM10 1

X18 OBJ 0.32162699 LIM1 1

X18 LIM11 1

X19 OBJ 0.24086335 LIM1 1

X19 LIM12 1

X21 OBJ 0.24872956 LIM2 1

X21 LIM4 1

X22 OBJ 0.18830030 LIM2 1

X22 LIM5 1

X23 OBJ 0.45251310 LIM2 1

X23 LIM6 1

X24 OBJ 0.18830030 LIM2 1

X24 LIM7 1

X25 OBJ 0.45251310 LIM2 1

X25 LIM8 1

X26 OBJ -0.18480703 LIM2 1

X26 LIM9 1

X27 OBJ 0.091772422 LIM2 1

X27 LIM10 1

X28 OBJ 0.24430312 LIM2 1

X28 LIM11 1

X29 OBJ 0.16167119 LIM2 1

X29 LIM12 1

X31 OBJ 0.23486331 LIM3 1

X31 LIM4 1

X32 OBJ 0.31007376 LIM3 1

X32 LIM5 1

X33 OBJ 0.23486331 LIM3 1

X33 LIM6 1

X34 OBJ 0.31007376 LIM3 1

X34 LIM7 1

X35 OBJ 0.23486331 LIM3 1

X35 LIM8 1

X36 OBJ -0.072083220 LIM3 1

X36 LIM9 1

X37 OBJ 0.32012302 LIM3 1

X37 LIM10 1

X38 OBJ 0.31949946 LIM3 1

X38 LIM11 1

X39 OBJ 0.23486331 LIM3 1

X39 LIM12 1

MARK0001 'MARKER' 'INTEND'

RHS

RHS LIM1 3

RHS LIM2 3

RHS LIM3 3

RHS LIM4 1

RHS LIM5 1

RHS LIM6 1

RHS LIM7 1

RHS LIM8 1

RHS LIM9 1

RHS LIM10 1

RHS LIM11 1

RHS LIM12 1

BOUNDS

UP bnd X11 1

UP bnd X12 1

UP bnd X13 1

UP bnd X14 1

UP bnd X15 1

UP bnd X16 1

UP bnd X17 1

UP bnd X18 1

UP bnd X19 1

UP bnd X21 1

UP bnd X22 1

UP bnd X23 1

UP bnd X24 1

UP bnd X25 1

UP bnd X26 1

UP bnd X27 1

UP bnd X28 1

UP bnd X29 1

UP bnd X31 1

UP bnd X32 1

UP bnd X33 1

UP bnd X34 1

UP bnd X35 1

UP bnd X36 1

UP bnd X37 1

UP bnd X38 1

UP bnd X39 1

ENDATA

後は、Shellで読み込んでやる。

SCIP> **read people\_allocation.mps**

...

SCIP> **optimize**

...

SCIP> **display solution**

Objective value: 2.75626088662

X38 1 (obj:0.31949946)

X12 1 (obj:0.36423028)

X14 1 (obj:0.36423028)

X16 1 (obj:-0.00044122338)

X21 1 (obj:0.24872956)

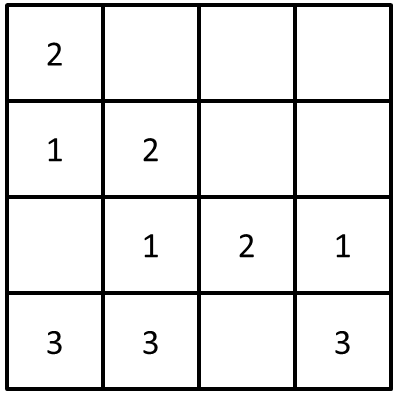
X23 1 (obj:0.4525131)

X25 1 (obj:0.4525131)

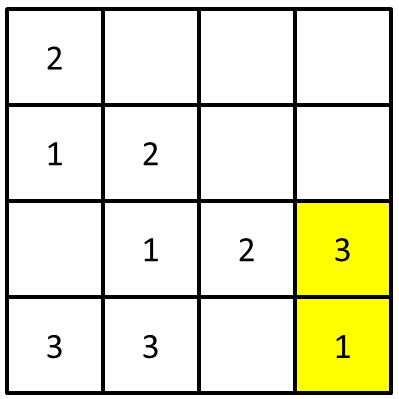
X37 1 (obj:0.32012302)

X39 1 (obj:0.23486331)

実際のセルで図示すると、以下のような結果だ。



一方、俺のgreedy アルゴリズムの結果は、以下の通り。まぁ、ほぼ一緒だけど、やはりoptimalではない。スコアは、2.69061893で、optimalと比べて約3%の誤差だ。



# アプリケーションとの連携

残念ながら、今のところコンパイル済みのライブラリファイルが見つからない。ギブアップ。

一般的に、linear programmingでＯＫな場合が多いので、とりあえずlinear programmingで解いちゃおう。Linear programmingなら、「lp\_solv」ライブラリが使える！