

クレジット:

UTokyo Online Education 数理手法Ⅲ 2018 寒野善博

ライセンス:

利用者は、本講義資料を、教育的な目的に限ってページ単位で利用することができます。特に記載のない限り、本講義資料はページ単位でクリエイティブ・コモンズ 表示-非営利-改変禁止 ライセンスの下に提供されています。

<http://creativecommons.org/licenses/by-nc-nd/4.0/>

本講義資料内には、東京大学が第三者より許諾を得て利用している画像等や、各種ライセンスによって提供されている画像等が含まれています。個々の画像等を本講義資料から切り離して利用することはできません。個々の画像等の利用については、それぞれの権利者の定めるところに従ってください。



整数計画のソルバーと LP ファイルによる問題の記述法

この資料では、汎用のソルバーを用いて線形計画問題や整数計画問題を解くための初歩的な方法について解説する。

1. ソルバー

線形計画や整数計画には、商用・非商用ともいくつかのソルバーがある（たとえば、web ページ [3] の “software” の項目では、最適化問題のタイプ別にソルバーの情報が集められている）。

たとえば、SCIP は非商業的な利用（アカデミック環境での利用）は無償であり、非商用ソルバーのなかでは最も高速なものの一つである^{*1}。また、GLPK も非商用（GNU GPL ライセンス）として手軽に使うことができるが、性能は SCIP に劣る^{*2}。IBM ILOG CPLEX は商用のソルバーであるが、IBM Academic Initiative Program への登録により非商業的な利用は無償になる^{*3}。同様に、MOSEK も商用であるが、アカデミックライセンスは無償で入手できる^{*4}。Gurobi Optimizer は商用のソルバーである（アカデミック環境では、6ヶ月間限定の無償アカデミックライセンスが利用できる）^{*5}。

Web ページ [3] の “benchmarks” の項目では、上記のものを含むさまざまな最適化ソルバーの性能を比較するベンチマーク実験の結果が継続的に公開されている。整数計画のモデリングやソルバーに関する情報は、文献 [1, 4] などでも得ることができる。

2. LP ファイル形式

最適化問題をソルバーに入力するには、いろいろな方法がある。ここでは、上記のソルバーが対応している LP ファイル形式の文法の基礎を解説する。LP ファイルとは、拡張子が `.lp` のテキスト形式のファイルであり、最適化問題を記述してソルバーに入力するために用いるものである。目的関数や制約をほぼ数式の通りに記述する形式であるので、理解しやすい。

次のページでは、まず、線形計画問題の簡単な具体例を LP ファイル形式で記述してみる^{*6}。

^{*1}ダウンロードページ：<http://scip.zib.de/>

^{*2}ダウンロードページ：<https://www.gnu.org/software/glpk/>

^{*3}ダウンロードページ：<http://www.ibm.com/software/integration/optimization/cplex/>

^{*4}ダウンロードページ：<https://www.mosek.com/>

^{*5}ダウンロードページ：<http://www.gurobi.com/>

^{*6}この資料に記載の LP ファイルは、講義資料の web ページ <https://www.or.mist.i.u-tokyo.ac.jp/kanno/lecture/> からダウンロードできる。

2.1. 簡単な例

例として、2変数の線形計画問題^{*7}

$$\text{Maximize } 70x_1 + 30x_2 \quad (1a)$$

$$\text{subject to } 5x_1 + 2x_2 \leq 80, \quad (1b)$$

$$x_1 + 3x_2 \leq 40, \quad (1c)$$

$$3x_1 + 7x_2 \leq 70, \quad (1d)$$

$$x_1 \geq 0, \quad x_2 \geq 0 \quad (1e)$$

をLPファイル形式で記述すると、次のようになる：

ファイル名：'product1.lp'

```
1 \ ( \ のあとは行の終わりまでコメント)
2 Maximize \ 最大化問題であることを示す.
3 obj: 70 x_1 + 30 x_2 \ 目的関数を書く.
4 \ 空行があってもよい.
5 Subject to \ ここから制約条件の記述が始まる.
6 5 x_1 + 2 x_2 <= 80 \ 最初の制約式を書く.
7 x_1 + 3 x_2 <= 40 \ 係数 1 は省略してもよい.
8 3 x_1
9 + 7 x_2 <= 70 \ 一つの制約式が複数行にまたがってもよい.
10
11 Bounds \ ここから変数の上下限値の記述が始まる.
12 0 <= x_1
13 0 <= x_2
14
15 End \ ファイルの終わりを表す.
```

2.2. 実行方法

SCIP を実行するには、SCIP のコマンドウィンドウで次のようにタイプする：

```
SCIP> read product1.lp
SCIP> optimize
SCIP> display solution
SCIP> write solution product1_scip.sol
```

すると、最適解が画面に表示され、ファイル 'product1_scip.sol' には次のように保存される：

'product1_scip.sol'

```
1 solution status: optimal solution found
2 objective value: 1127.58620689655
3 x_1 14.4827586206897 (obj:70)
4 x_2 3.79310344827586 (obj:30)
```

^{*7}この問題は、生産計画問題の具体例である。講義資料『CVXPY による線形計画問題の解き方』を参照のこと。

CPLEX の使い方もほぼ同じで、CPLEX のコマンドウィンドウで次のようにタイプする：

```
CPLEX> read product1.lp
CPLEX> optimize
CPLEX> display solution variable -
CPLEX> write product1_cplex.sol
```

これで、最適解が画面に表示される。また、ファイル ‘product1_cplex.sol’ には最適解の情報が xml 形式で保存される。

GLPK を使う場合は、コマンドプロンプトで

```
> glpsol --cpxlp product1.lp -o solution.txt
```

のようにタイプする。

2.3. LP ファイル形式の解説

- バックスラッシュ (\) の右側はコメント扱いになる。
- 目的関数のセクション
 - 最小化問題では ‘Minimize’, 最大化問題では ‘Maximize’ で書き始める（大文字・小文字の別は任意）。
 - 変数名は、16 文字を超えない範囲で、アルファベットと数字を組み合わせる自由につけることができる。
 - 演算子と変数の間や、係数と変数の間には、半角スペースを入れる。目的関数は複数行にまたがってもよい。ただし、一つの変数名の途中で改行を入れてはいけない。
- 制約のセクション
 - ‘Subject to’ に続いて、制約式を書き並べる。
 - それぞれの制約式には、名前をつけることもできる。その場合、制約式の左に ‘名前:’ とタイプする。
 - 一つの制約式は、新しい行から書き始める。
 - 制約式は、等号 ‘=’ か不等号 ‘<=’, ‘>=’ を用いて記述する。変数を含む項は、等号や不等号の左辺にまとめる。変数を含まない項（つまり、定数項）は、等号や不等号の右に書く。
- 変数の上下限值セクション
 - ‘Bounds’ に続いて、それぞれの変数の上下限值制約を書き並べる。
 - 上限値と下限値の両方を指定するときは、たとえば ‘2 <= x_1 <= 3.5’ などと書く。
 - このセクションに登場しない変数は、非負変数であるとみなされる（従って、2.1 節の例では、実は、‘0 <= x_1’ と ‘0 <= x_2’ は書かなくてもよい）。
 - 自由変数（上下限值がない変数）は、‘-inf <= x_2 <= inf’ と書くか、‘x_2 free’ と書く（このように書いておかないと、自動的に ‘0 <= x_2’ という制約が課されてしまう）。

- 変数型のセクション
 - 問題に整数変数が含まれるときには、変数型を指定する（2.1節の例は線形計画問題なので、このセクションはない）。
 - ‘Binary’ に続いて、0-1 変数の変数名を並べる。
 - ‘General’ に続いて、整数変数の変数名を並べる。なお、負の整数値も許す変数の場合には、「変数の上下限值セクション」でそのように指定しておく必要がある（その指定をしないと、非負の整数変数とみなされる）。
- ファイルの最後には ‘End’ と書く。
- LP ファイル形式の詳細は、たとえば web 上の CPLEX のマニュアル [2] などに情報がある。ただし、ソルバーによっては細かい書式が少し異なる場合がある。

2.4. 混合整数計画の例

実数変数 x_1, x_2, x_3 と整数変数 x_4 をもつ混合整数計画問題

$$\begin{aligned}
 &\text{Maximize} && x_1 + 2x_2 + 3x_3 + 4x_4 \\
 &\text{subject to} && -x_1 + x_2 + x_3 + 10x_4 \leq 20, \\
 &&& x_1 - 3x_2 + x_3 \leq 30, \\
 &&& x_2 - 3.5x_4 = 0, \\
 &&& 0 \leq x_1 \leq 40, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad 2 \leq x_4 \leq 3, \\
 &&& x_4 \text{ は整数}
 \end{aligned}$$

を LP 形式で記述すると、次のようになる：

ファイル名：‘mip_example.lp’

```

1 Maximize
2  obj: x1 + 2 x2 + 3 x3 + x4
3 Subject to
4  c1: - x1 + x2 + x3 + 10 x4 <= 20
5  c2: x1 - 3 x2 + x3 <= 30
6  c3: x2 - 3.5 x4 = 0
7 Bounds
8  0 <= x1 <= 40
9  2 <= x4 <= 3
10 General
11  x4
12 End

```

なお、最適解は

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 40 \\ 10.5 \\ 19.5 \\ 3 \end{bmatrix}$$

である。

参考文献

- [1] 藤江 哲也：整数計画法による定式化入門. オペレーションズ・リサーチ, **57**, 190–197 (2012).
- [2] IBM ILOG: *IBM ILOG CPLEX Optimization Studio Documentation*. <http://www.ibm.com/support/knowledgecenter/> (Accessed July 2018).
- [3] H. Mittelmann: *Decision Tree for Optimization Software*. <http://plato.asu.edu/guide.html> (Accessed August 2018).
- [4] 宮代 隆平：整数計画ソルバー入門. オペレーションズ・リサーチ, **57**, 183–189 (2012).

(以上)