

都市情報システム実習（檀担当分） [第 1 週]

1 本実習の概要

1.1 概要

- 本実習では最適化（数理計画法）に関する実習を行います。
- 『オペレーションズ・リサーチ 1』という科目で最適化（特に線形計画問題）の理論的な基礎を勉強しましたが、本実習ではソフトウェアを用いて実際の問題を解く練習を行います。
- また、上記実習を通じて、『充実した』レポートを書く練習を行います。

1.2 各週計画（予定）

- 第 1 週：最適化ソルバー glpk の利用、基本的な問題に対する求解
- 第 2 週：さまざまなモデル構築の練習
- 第 3 週：巡回セールスマン問題の求解 (1)
- 第 4 週：巡回セールスマン問題の求解 (2)
- 第 5 週：個別面談、フィードバック、レポート修正

1.3 成績評価

- 成績評価の方法：実習時間内の成果（40 %）、レポート（60 %）
- 実習時間内の成果物・レポートとも第 4 週（※「第 5 週」ではない）終了後に提出してもらいます。毎回提出する必要はありません。
- 提出頂いたレポートに関する個別面談を第 5 週に行い、理解の度合い等を計ります。
- 【重要】他人のレポートの剽窃（カンニング）等があった場合、本科目の単位を出すことができませんので、そのようなことが『絶対に』ないようにしてください。
- カンニングについては以下のように扱います：
 - － 同じ班でのカンニング：カンニングをした側・された側の両方について、檀担当分の配点を 0 点にします

- － 違う班でのカンニング：（時間的に後ということが明確ですので）後に実習を行った班に属する側について、檀担当分の配点を 0 点にします

1.4 レポートの作成方法

- PDF ファイルで作成してください（Word で作成して PDF 化するのが便利かもしれませんが、その他、PDF であれば何で作成したものでも OK です）。
- 実習の課題は少なめにしているつもりです。その分、（実習時間中を含めて、）レポートの記述を充実させてください。結果を貼り付けただけのレポートなど、論外です。実験手順、結果、考察等、レポートとして必要な項目を自分で考えて（調べて）、できるだけよいレポートを作成してください。
- 【重要】レポートには、自分が理解していないことを書かないでください（課題が全て終わってなくても構いません。減点はしますが）。個別面談では、レポートに記載している内容について質問する予定です（課題が全て終わっていないことより、レポートに記載している内容が理解できていないことの方がより悪いことです）。

2 最適化問題とソルバー

2.1 最適化問題（数理計画問題）

最適化問題（数理計画問題）は、一般的に次のように表現されます：

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathcal{S} \end{array} \quad (1)$$

ここで \mathcal{S} は「実行可能領域」といい、変数 \mathbf{x} が値を取り得る領域全体を指します。また $f(\mathbf{x})$ は最小化¹すべき関数で「目的関数」といいます。つまり、最適化問題とは

実行可能領域の中で、目的関数を最小化するような点を探す

¹問題によっては「最大化 (maximize)」になることもあります。ただしその場合でも、「 f を最大化」 \Leftrightarrow 「 $-f$ を最小化」なので、(1) の形に帰着することができます。

問題であると言えます。

最適化問題は、目的関数 f と実行可能領域 S を定める制約条件の数学的性質により分類することができます。例えば、変数が全て連続変数（実数値を取るような変数）である場合の問題のクラス分けは表 1 の通りです²。

表 1: 最適化問題の分類（連続変数の場合）

f	S	問題クラス
線形	線形	線形計画問題 (LP)
(凸) 二次	線形	(凸) 二次計画問題 (QP)
少なくとも一方が非線形	非線形	非線形計画問題 (NLP)

一方、最適化問題の変数の一部ないし全部が整数変数（整数値しか取らないような変数）である場合もあります。その場合は、上の問題クラス名の先頭に「混合整数 (Mixed Integer: 'MI')」をつけると、該当する問題のクラス名になります：「混合整数線形計画問題 (MILP)」, 「混合整数 (凸) 二次計画問題 (MIQP)」, 「混合整数非線形計画問題 (MINLP)」。

この実習では、主に LP, MILP について実習を行います。

2.2 ソルバー

実際の最適化問題を解くときには、最適化問題を解くためのソルバーを利用するのが一般的です。商用・非商用を問わず様々なソルバーがありますが、この実習では 'glpk' というソルバーを使います。

3 実習 1 : glpk を動かしてみる

それではまず、簡単な問題を用いて glpk を動かしてみましょう。

次の問題を解いてみたいと思います [1, p.30] :

$$\begin{array}{ll} \text{minimize} & -x_1 - x_2 \\ \text{subject to} & 3x_1 + 2x_2 \leq 12 \\ & x_1 + 2x_2 \leq 8 \\ & x_1 \geq 0, x_2 \geq 0 \end{array} \quad (2)$$

'glpk'（やその他の最適化ソルバーの大半）はシンプレックス法を用いて問題を解きますが、それらを用いる際には問題をそのまま入力すれば OK です。

この問題を解くためには、まず、問題を定義するファイルを作る必要があります。'w1p1.mod' (week 1, problem 1 の意味) という名前で次の内容のファイル

²この他にも様々な問題クラスがあります。

を作ってください：

```
var x1 >= 0;
var x2 >= 0;

minimize Objective: - x1 - x2;

subject to C1: 3 * x1 + 2 * x2 <= 12;
subject to C2: x1 + 2 * x2 <= 8;
```

問題の記法は直感的にわかると思います（追って、様々な記法を勉強していくことになります）。

こうやって作ったファイルを glpk に「喰わせる（＝読み込ませる）」ことによって、問題を解くことができます。それでは、LMS から glpsol.exe をダウンロードして、実際に問題を解いてみましょう。

コマンドプロンプトで、glpsol.exe が保存されているフォルダに移動し（'cd' コマンドを使います。過去の様々な実習でやっているはずですよ？）、

```
> glpsol.exe
```

としてみてください。すると、

```
No input problem file specified;
try glpsol.exe --help
```

と表示されると思います。「問題ファイルが指定されていないよ。glpsol.exe --help を試してみてね」ということですね³。それでは glpsol.exe --help を試してみると、たくさんが表示がされます。これは、glpsol.exe というコマンド（ソルバー）のオプションを表示しています。（少なくともこの実習では）すべてを知る必要はありませんが、-m, -d, -o, --log オプションを用いることになりますので（今後増えるかも）、それらのオプションに何かが書いてあるか、読んでみて下さい。

それでは、オプションの意味を理解し、'w1p1.mod' を 'glpsol.exe' と同じフォルダに配置した上で、

```
> glpsol.exe -m w1p1.mod -o w1p1_result.txt
--log w1p1_log.txt
```

と実行してみてください（※実際は 1 行で書く）。すると、(問題ファイルを正しく作成していれば、) 'w1p1_result.txt', 'w1p1_log.txt' というファイルが生成されていると思います。前者が解ファイル（求解結果が書いてある）、後者がログファイル（求解時のメッセージが書いてある：コマンドプロンプトに表示されるのと同じもの）です。解ファイルを見てみると、

³余談ですが、エラーメッセージはちゃんと読みましょう。英語だからということでも嫌いしている人も多いみたいですが...

No.	Column name	St	Activity
1	x1	B	2
2	x2	B	3

と書いてある箇所があると思います（一部省略）。これは、変数 x_1 の値が 2, 変数 x_2 の値が 3 であることを示しています。これは [1] にある答えと（もちろん）一致します。

4 実習 2: 添字を持つ問題の求解, データファイルの記述

次に、以下の「輸送問題」[2] をソルバーを用いて解いてみたいと思います：

あるメーカーの 3 カ所の工場 A, B, C の供給量 a_i , および、4 カ所の都市の需要量 b_j , そして各工場 i から各都市 j へ運ぶのにかかる 1 単位当たり輸送費用 c_{ij} と該当経路上の輸送上限 u_{ij} が、表 2, 3 のように与えられているとする。このとき、工場の供給上限、都市の需要、各経路の輸送上限を満たし、かつ、輸送費用の合計を最小にする輸送計画を求めよ。

表 2: 輸送費用 c_{ij}

	都市 1	2	3	4
工場 A	4	1	3	3
B	9	2	7	10
C	8	1	10	9

表 3: 上限 u_{ij} , 供給 a_i , 需要 b_j

	都市 1	2	3	4	供給
工場 A	28	25	21	28	60
B	17	15	15	25	45
C	22	17	19	10	35
需要	35	15	35	45	—

この輸送問題を定式化すると、次のようになります：

$$\begin{aligned}
 & \text{minimize} && \sum_{i \in I, j \in J} c_{ij} x_{ij} \\
 & \text{subject to} && \sum_{j \in J} x_{ij} \leq a_i \quad (i \in I) \\
 & && \sum_{i \in I} x_{ij} = b_j \quad (j \in J) \\
 & && 0 \leq x_{ij} \leq u_{ij} \quad (i \in I, j \in J)
 \end{aligned} \tag{3}$$

ここで、 I は工場の集合 $\{A, B, C\}$ を表し、 J は都市の集合 $\{1, 2, 3, 4\}$ を表します。

さて、この問題を glpk を用いて解くためには、次の 2 つのことを学習する必要があります：

- 添字を持つ問題の求解
- データファイルの記述

以下、これらのことについて説明します。

問題 (3) を glpk が読むことができる問題ファイルとするためには、次のように書く必要があります（ファイル名は wlp2.mod などとするとよいでしょう）：

```

set I;
set J;

var x{I, J} >= 0;

param c{I, J};
param u{I, J};
param a{I};
param b{J};

minimize Objective:
    sum {i in I, j in J} (c[i, j] * x[i, j]);

subject to C1 {i in I}:
    sum {j in J} x[i, j] <= a[i];

subject to C2 {j in J}:
    sum {i in I} x[i, j] == b[j];

subject to C3 {i in I, j in J}:
    x[i, j] <= u[i, j];

```

これも、概ね直感的に記述内容はわかると思います。注意すべき点としては、添字を持つ変数・パラメータの宣言時には '{ }' を、数式の中では '[]' を利用すること、でしょうか。

ただし、ここには、集合の要素やパラメータの具体的な値は書かれていません。これは、別途パラメータファイル (wlp2.dat) を作成し、そちらに記述します。記述の方法は以下ようになります：

```

set I := A B C;
set J := 1 2 3 4;

param c : 1 2 3 4 :=
A 4 1 3 3
B 9 2 7 10
C 8 1 10 9
;

param u : 1 2 3 4 :=
A 28 25 21 28

```

```

B 17 15 15 25
C 22 17 19 10
;

param a :=
A 60
B 45
C 35
;

```

```

param b :=
1 35
2 15
3 35
4 45
;

```

':=' や ':' の使いどころ（後者は 2 次元データのように用いている）に注意すれば、これも概ね直感的に理解できるはずです。

この 2 つのファイルを準備すれば、求解が可能になります。

```

> glpsol.exe -m wlp2.mod -d wlp2.dat
-o wlp2_result.txt --log wlp2_log.txt

```

とすれば実行できます（※実際は 1 行で書く）。データファイルがあるので '-d' オプションを用いていることに注意してください。

5 実習 3: 整数変数を持つ問題の求解

次は、以下の問題を解いてみましょう [2]：

某大学では年々増加する学術雑誌の購読料高騰に頭を悩ませている。来年度は予算もカットされることから、これまで購読してきた雑誌 a~r のうちいくつかの雑誌の購読料を中止し、数を絞って購読することが避けられない。そこで、所属教員にアンケートを募り、その希望度及び価格をまとめたのが表 4 である。予算が 100 であるとき、希望度合計を最大にするよう購読継続雑誌を選定せよ。

これは、いわゆる「ナップサック問題」[1, p.156] と呼ばれる問題です。この問題を定式化すると、次のようになります：

$$\begin{array}{ll}
 \text{maximize} & \sum_{i \in I} b_i x_i \\
 \text{subject to} & \sum_{i \in I} c_i x_i \leq 100 \\
 & x_i \in \{0, 1\} \quad (i \in I)
 \end{array} \quad (4)$$

表 4: 雑誌の価格と購読希望度

雑誌 価格 希望度	a	b	c	d	e	f	g	h	i
	2	13	13	15	20	10	12	3	10
	9	3	3	1	10	7	10	0	1
雑誌 価格 希望度	j	k	l	m	n	o	p	q	r
	7	5	8	10	5	5	9	5	4
	1	6	3	5	0	2	15	3	0

ここで、 b_i は雑誌 i の希望度、 c_i は雑誌 i の価格を表すパラメータ、 x_i は雑誌 i の購読の有無を表す 0-1 変数です。注意すべき点は、変数が連続変数ではなく、整数変数、特に 0-1 変数になっている、という点です。

この問題を glpk を用いて解いて下さい。前節と同じようなやり方をすればよいと思います。

【ヒント】

- 'minimize' の代わりに 'maximize' を用いることができます（※目的関数を (-1) 倍して minimize するという方法も考えられますが、その必要はありません。）

- 変数の宣言時に 'binary' とすると、0-1 変数として定義されます。

【例】var x{I} binary;

6 レポートについて

本日の実習課題は以上です。時間的にはかなり余裕があると思っています。その分、レポートの記述を充実させて欲しいという願いから、そのようにしています。

これまでのいくつかの科目における実験レポートを見てきましたが、レポートとしての体裁を満たしているものは、残念ながら非常に少なかったと言わざるを得ません。ですので、この実習（檀担当分）では、実習時間中にもレポートを書く時間をできるだけ取りたいと思っています。また、(最初に書いたように) そのレポートに対する（口頭試問を通じた）フィードバックをすることで、最終版が充実したレポートになることを期待しています。

それゆえ、自分で内容を理解していないことを書いていたり、他人のレポートを写しているようなレポートについては厳しい評価をしたいと思っています（本気です！）。課題が全部できていないことは（減点はするかもしれませんが）決定的に悪いことではありませんので、他人のレポートを写したりするようなことは決しないように。

何を書けばいいのか、ということについても、自分で考えたり調べたりしてみましょう。こちらで特定の形式は決めることはしません。レポートの書き方に関する本は図書館にいくらでもあると思いますし、本屋

さんにもたくさんあるはずです。そういったものを見ながら、自分なりのレポートを（本気で！）仕上げて下さい。

参考文献

- [1] 福島雅夫，新版 数理計画入門，朝倉書店 (2011).
- [2] 後藤順哉，Excel で始める数理最適化，日本オペレーションズ・リサーチ学会学会誌，平成 24 年 4 月号 (2012).