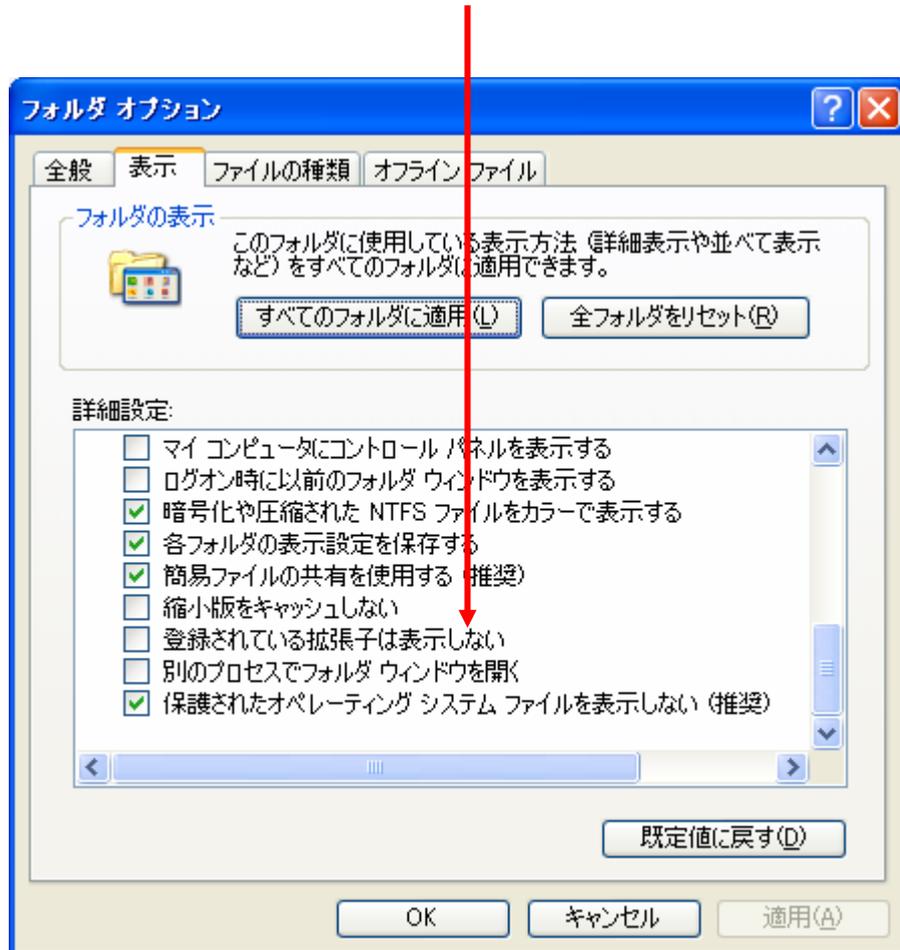


AMPL 例題（ナップサック問題の連続緩和）

椎名 孝之

1. はじめに

AMPL(A Modeling Language for Mathematical Programming)をインストールした `amplcm1` フォルダには、`ampl.exe` という AMPL の実行ファイルがあることを確認する。また、適当なフォルダを開いて、「ツール」→「オプション」をクリックし、次のフォルダオプションのウィンドウを開く。ここで、「登録されている拡張子は表示しない」のチェックを外す。元々外してあれば不要です。



このように拡張子を表示することで、AMPL で使用するファイルの識別が容易になる。

AMPL を使用するとき、次のようなファイルの操作を必要とする。

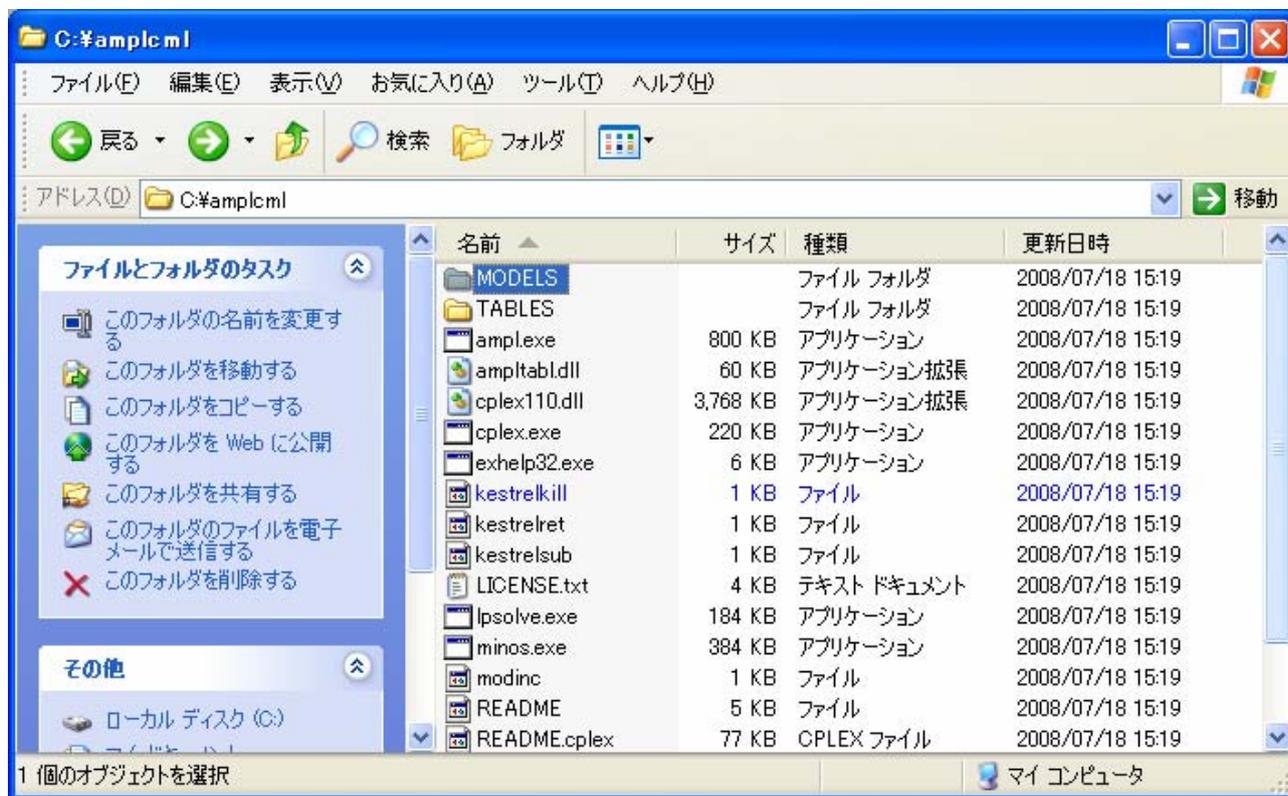
`.mod` ファイル： 数理計画モデルを記述したファイル

`.dat` ファイル： 数理計画モデルにおけるデータを記述したファイル

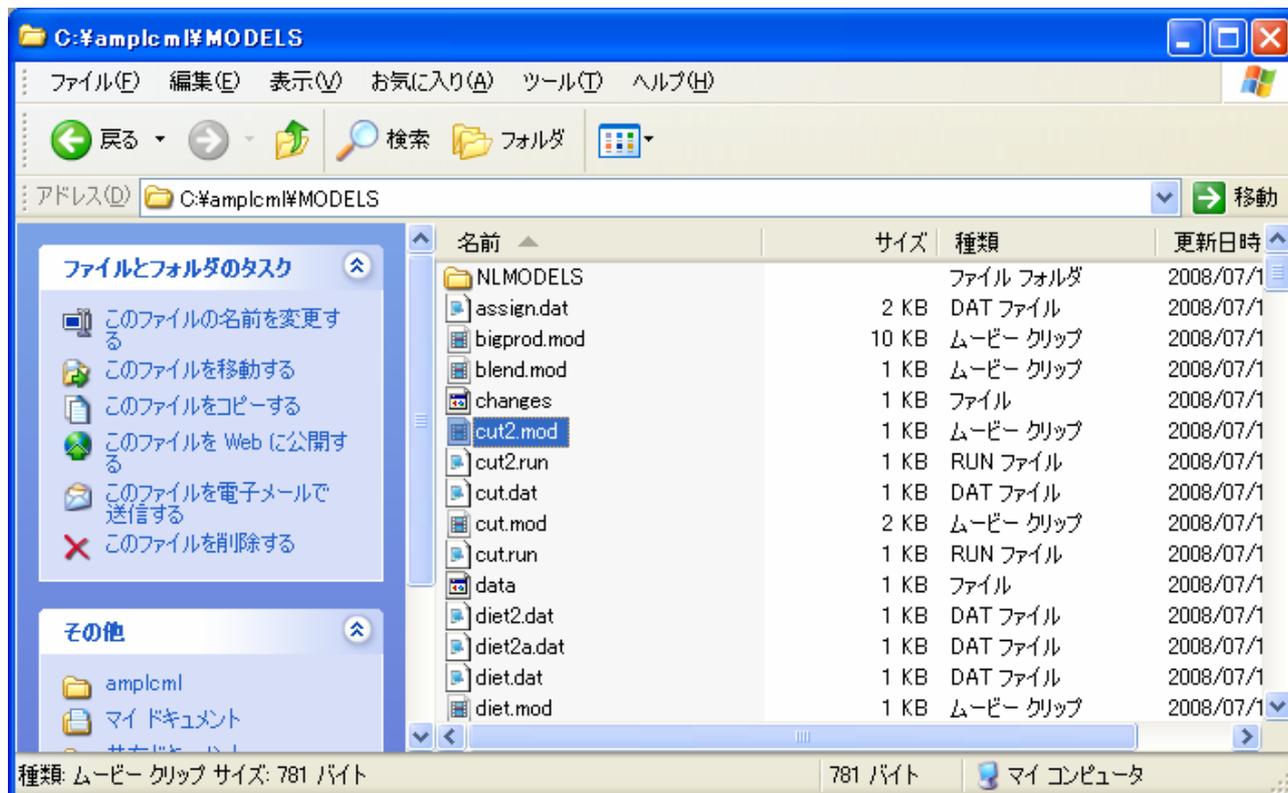
`.run` ファイル： AMPL のコマンドを記述したファイル

これらのファイルは、メモ帳 (`notepad`) あるいはワードパッドなどのテキストエディタで操作できるようにファイルとの関連付けを行う。AMPL がインストールされたフォルダ `C:\¥amplcm1` の中には、`MODELS` と `TABLES` の2つのフォルダが存在する（確認されたい）。フォルダ `MODELS` には、数理計画モデルファイル (`.mod`) のファイルが保存されている。これらのファイルは AMPL 解説書「AMPL: A Modeling Language for Mathematical Programming」(by Robert Fourer, David M. Gay, and Brian

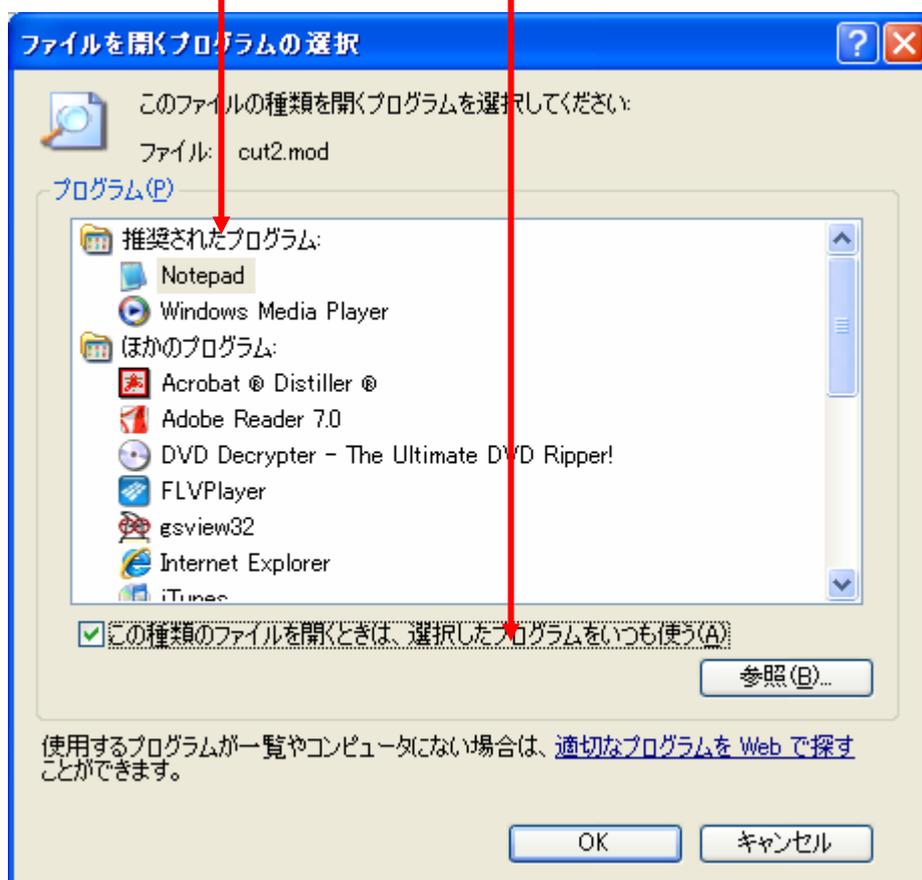
W. Kernighan, Duxbury Press / Brooks/Cole Publishing Company, 2002. ISBN 0-534-38809-4) に掲載されているものと同一である。



フォルダ MODELS の中を見てみよう。



ここには、上に示した以下の .mod ファイル、.dat ファイル、.run ファイルの多くの例題がある。これらのファイルはAMPLで記述したテキストファイルであるが、WindowsXPなどの標準的な環境では、必ずしもテキストエディタとは関連付けられてはいない。例えば、.mod ファイルはムービークリップとして、Windows media player と関連付けられている（しかし、メディアプレーヤーで.mod ファイルを開いても当然ながら再生できない）。そのため、適当な.modファイルを選び、右クリックして「プログラムから開く (H)」をクリックし、「プログラムの選択 (C)」を左クリックする。「ファイルを開くプログラムの選択」ウインドウにおいて、「推奨されたプログラム」あるいは「ほかのプログラム」の中からNotepadを選択し、「この種類のファイルを開くときは、選択したプログラムをいつも使う (A)」をチェックする。



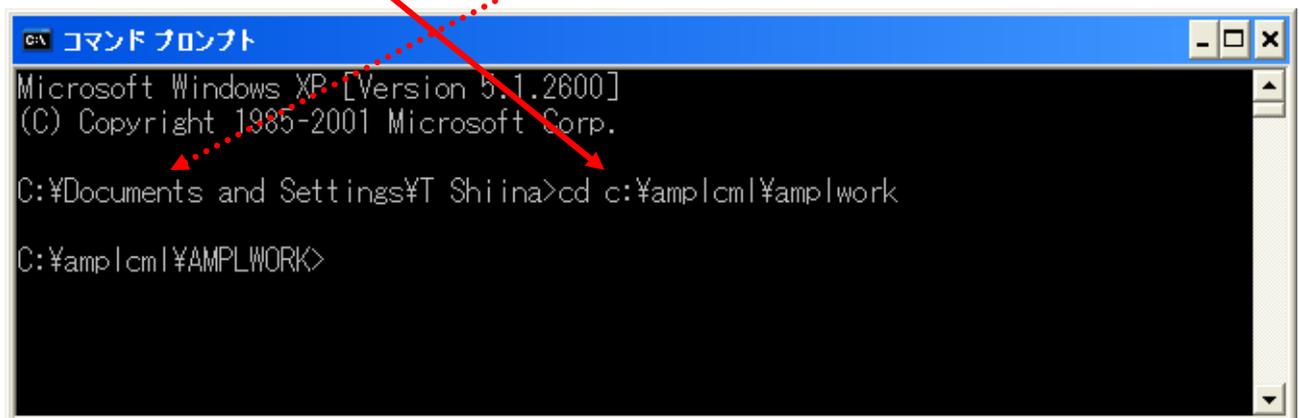
これで.mod ファイルをメモ帳 (Notepad) で開くことが可能になった。同様の操作を繰り返すことによって、.dat ファイルと.run ファイルをメモ帳 (Notepad) で開くことができることを確認されたい。

2. 作業フォルダの準備

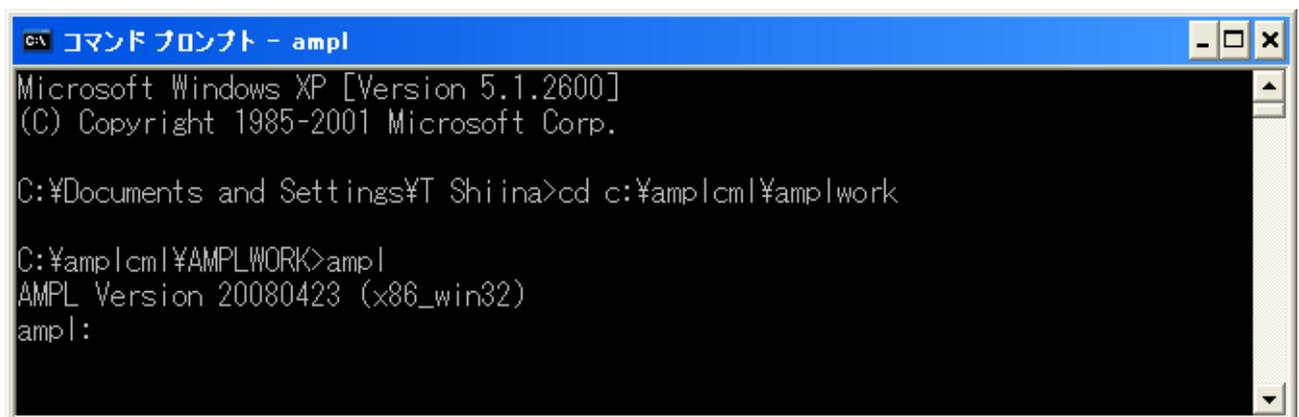
フォルダ `amplcml` の下にフォルダ `AMPLWORK` を作成する。



新規作成されたこのフォルダAMPLWORKを、コマンドプロンプトで開く。コマンドプロンプト（スタート→すべてのプログラム→アクセサリ）では通常マイドキュメントのフォルダが開くため、「cd c:\%amp|cml\%amp|work」で AMPLWORK フォルダに移動する。



このフォルダでも AMPL が起動できることを確認されたい。Ctrl+c で終了できる。



3. 線形計画問題に対する適用例

次のような 3 つのファイルを作業フォルダ AMPLWORK (C:\¥amplcml¥amplwork) に作成する。

knapsack-relax.mod ファイル：ナップサック問題の連続緩和問題を記述したファイル

knapsack-relax.dat ファイル：ナップサック問題のデータを記述したファイル

knapsack-run ファイル：AMPL のコマンドを記述したファイル

以下に内容を記す。ただし「#」以降はコメントである。

ファイル knapsack-relax.mod の内容

```
param Nitem;           #品物数を Nitem と定義
param weight{i in 1..Nitem}; #各品物に重さ (weight) を定義
param value{i in 1..Nitem}; #各品物に価値 (value) を定義
param l{i in 1..Nitem};  #0-1 制約の緩和制約における下限
param u{i in 1..Nitem};  #0-1 制約の緩和制約における上限
param capacity;         #ナップサックの容量
var x{i in 1..Nitem} >=0;      #各品物 (入れる/入れない) に対応する変数
maximize Profit: sum{i in 1..Nitem}value[i]*x[i]; #目的関数 Profit
subject to Capacity_Constraint: sum{i in 1..Nitem}weight[i]*x[i] <= capacity;
                                #ナップサックの容量制約
subject to Upper_Lower_Bound {i in 1..Nitem}: l[i] <= x[i] <=u[i];
                                #0-1 制約を緩和した上下限
```

ファイル knapsack-relax.dat の内容

```
param Nitem:= 3;
param value :=
1 7
2 8
3 3;
param weight :=
1 3
2 4
3 2;
param capacity:=
6;
param l :=
1 0
2 0
3 0;
param u :=
1 1
2 1
3 1
;
```

ファイル knapsack-relax.run の内容

```
model knapsack-relax.mod ;
data knapsack-relax.dat ;
option display_round 6; #小数点以下 6 桁表示
option solver cplex;      #ソルバに CPLEX を指定、これがないと MINOS が動く
solve;                    #問題の求解
expand > knapsack-relax.sol;      #ファイル(knapsack-relax.sol)に定式化表示
display Profit > knapsack-relax.sol; #ファイル(knapsack-relax.sol)に目的関数値
display x > knapsack-relax.sol;    #ファイル(knapsack-relax.sol)に最適解
display Capacity_Constraint.dual > knapsack-relax.sol; #ナップサック制約の双対変数
display Upper_Lower_Bound.dual > knapsack-relax.sol; #0-1 緩和制約の双対変数
```

4. ampl の実行

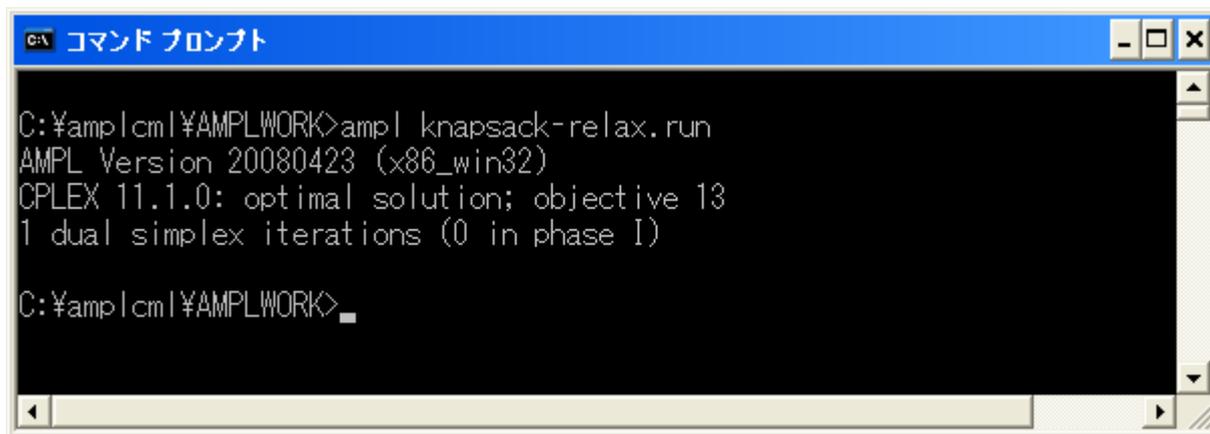
コマンドプロンプトを開き、 amplwork フォルダ (C:¥amp|cm|¥amplwork)において、

「**ampl knapsack-relax.run**」

と入力する。正常に起動し終了すると、次のように

C:¥amp|cm|¥AMPLWORK>

と表示され終了する。



```
C:\¥amp|cm|¥AMPLWORK>ampl knapsack-relax.run
AMPL Version 20080423 (x86_win32)
CPLEX 11.1.0: optimal solution; objective 13
1 dual simplex iterations (0 in phase I)

C:\¥amp|cm|¥AMPLWORK>
```

5. 実行結果の確認

ファイル knapsack-relax.sol の内容を示す。ただし.sol ファイルについても、テキストエディタから開けるようにプログラムとの関連付けを行っておくことを忘れないように。

maximize Profit:

$$7*x[1] + 8*x[2] + 3*x[3];$$

subject to Capacity_Constraint:

$$3*x[1] + 4*x[2] + 2*x[3] \leq 6;$$

subject to Upper_Lower_Bound[1]:

$$0 \leq x[1] \leq 1;$$

subject to Upper_Lower_Bound[2]:

$$0 \leq x[2] \leq 1;$$

subject to Upper_Lower_Bound[3]:

$$0 \leq x[3] \leq 1;$$

Profit = 13.000000

x [*] :=

1 1.000000

2 0.750000

3 0.000000

;

Capacity_Constraint.dual = 2.000000

Upper_Lower_Bound.dual [*] :=

1 1.000000

2 0.000000

3 0.000000

;

以上