

## Python 版 数理計画モデリング言語 pysimple のご紹介

NTT データ数理システム 池田悠<sup>1</sup>

### 1 はじめに

近年, 機械学習分野への注目が高まる中, 最適化(以下, 数理計画問題)についても再評価がされつつある. 計算機から機械学習のフレームワークを扱うインターフェースはプログラミング言語 Python が主流となっており, 数理計画問題についても Python から直接扱うための, より直感的なモデリング言語が望まれている.

NTT データ数理システムで開発している数理計画法パッケージ Numerical Optimizer<sup>2</sup> では専用のモデリング言語として SIMPLE<sup>3</sup> が付属している. SIMPLE は C++ をベースにした言語である. また, R 言語でモデルを記述できる RNUOPT<sup>4</sup> も存在している. 本稿では Python 版 SIMPLE の開発について報告をする.

### 2 pysimple

pysimple は数理計画問題のためのモデリング言語であり, Numerical Optimizer のソルバーと接続することができる. Python のモジュールとして実装しているため, Python ユーザはスムーズに連携することができる. また, Numerical Optimizer の SIMPLE に近い文法を備えているため, SIMPLE ユーザ如何に関わらず, 数式をそのまま記述する感覚でモデリングを行うことができる. 以下は SIMPLE 例題集<sup>5</sup>の配合問題を pysimple でモデリングしたものである. ただし, データ渡し部分は省略している. また, インターフェースは開発中のものであり, 製品版では変わる可能性がある点に注意いただきたい.

```
from pysimple import *
p = Problem(name='配合問題')

Alloy = Set(value=range(1, 10), name='市販の合金集合')
i = Element(set=Alloy)
Blend = Set(value=('Lead', 'Zinc', 'Tin'), name='構成金属集合')
j = Element(set=Blend)

r = Parameter(index=(i,j), name='構成比率')
c = Parameter(index=i, name='コスト')
b = Parameter(index=j, name='目標比率')
x = Variable(index=i, lb=0, name='混合比率')

p += Sum(c[i]*x[i], i)      # 目的関数

p += Sum(x[i], i) == 1
p += Sum(r[i,j]*x[i], i) == b[j]

print(p)
status = p.solve()

print(p.objective)
print(p.x)
```

文法の多くは SIMPLE を踏襲しているため、異なる点を幾つか述べる。

まず、モデル中に問題クラス `Problem` を宣言し、目的関数、制約式を追加する方式とした。これにより複数の問題をモデル中で扱えるようになった。

次に、宣言時にデータを直接与えることができるようになった。上記のサンプルでは書かれていないが、`Set` の他にも

`Parameter` などに `value` 属性をつけて宣言することにより、Python で作成したデータをそのまま渡すことができる。

また、スパースモデリングもでき、多次元集合や多次元 `Element` を、より直感的に扱うことができる。以下は SIMPLE 例題集の割当問題を `pysimple` でモデリングしたものである。

```
from pysimple import *
p = Problem(name='割当問題')

JPPair = Set(dim=2) # Job(仕事)とPeople(人)の組合せ(疎)
jp = Element(set=JPPair)
x = BinaryVariable(index=jp)
cost = Parameter(index=jp, name='コスト')
jyukuren = Parameter(index=jp, name='熟練度')

# 各人に割り振る仕事は、最大で 3 つまでとする
p += Sum(x[jp], jp(1)) <= 3

# 発注、ごみ捨て、買出し は 1 人を割り当てる
jp1 = ElementOf(jp, jp(2)<['発注', 'ごみ捨て', '買出し'])
p += Sum(x[jp1], jp1(2)) == 1
:
```

SIMPLE と異なり、要素を限定した `Element` を先行して用意しておく必要があるが、モデルの構造自体はより明確になっている。

他にも演算に伴う `name` 属性自動付与、変数の実数緩和、などの便利な新機能が付属している。

### 3 まとめと今後の展望

Python で数理計画問題を扱うモデリング言語 `pysimple` を開発している。現在は線形計画問題しか扱うことができないが、今後は WCSP や二次計画問題を扱えるようにしたい。

---

<sup>1</sup> ikeday@msi.co.jp

<sup>2</sup> <http://www.msi.co.jp/nuopt/>

<sup>3</sup> <http://www.msi.co.jp/nuopt/products/simple.html>

<sup>4</sup> <http://www.msi.co.jp/nuopt/products/derivation/rnuopt/index.html>

<sup>5</sup> <http://www.msi.co.jp/nuopt/docs/v20/examples/html/01-00-00.html>