Python 🐍 and AMPL:
Build Prescriptive Analytics applications quickly with Pandas, Colab, Streamlit, and amplpy

Filipe Brandão, Robert Fourer
{filipe,4er@ampl.com}
AMPL Optimization Inc.
www.ampl.com – +1 773-336-AMPL

INFORMS 2023, Phoenix, Arizona — October 15-18, 2023, CC-North 231B, Technology Tutorial, Tuesday, October 17, 2:55 – 3:30 pm
Outline

---

**Part I:**
- Quick introduction to AMPL and new modeling functionalities

**Part II (live demos):**
- Quick introduction to amplpy (our Python API)
- AMPL on Google Colab
  - AMPL Model Colaboratory (https://colab.ampl.com)
- AMPL and solvers as python packages
- AMPL on Streamlit Cloud
- How to deploy large-scale optimization applications with AMPL
Formulating Models More Like You Think About Them

Describe an optimization problem
  ○ In a form *you find natural or convenient*
  ○ Using readily recognized expressions

Send it to a solver
  ○ In a form *the solver will accept*
  ○ Relying on the modeling software to translate

Get back a result
  ○ In the form *you originally used*
Thank you so much for replying.

Let me show my "if-then" constraint in a more clear way as follows:

set veh := {1..16 by 1};

param veh_ind {veh};
param theory_time {veh};
param UP := 400000;

var in_lane_veh {veh} integer >=1, <=2;
var in_in_time {veh} >=0, <=UP;

Note that "in_lane_veh {veh}" are integer variables which equal 1 or 2, and "in_in_time {veh}" are continuous variables.

subject to IfConstr {i in 1..card(veh)-1, j in i+1..card(veh):
    veh_ind[i] = veh_ind[j] and theory_time[i] <= theory_time[j]}:

When I run my program, there appears the following statement:

CPLEX 20.1.0.0: logical constraint _slogcon[1] is not an indicator constraint.
To reformulate this model in a way that your MIP solver would accept, you could define some more binary variables,

\[
\text{var in\_lane\_same \{veh,veh\} binary;}
\]

with the idea that \(\text{in\_lane\_same}[i,j]\) should be 1 if and only if \(\text{in\_lane\_veh}[i] = \text{in\_lane\_veh}[j]\).

Then the desired relation could be written as two constraints:

\[
\text{in\_lane\_veh}[i] = \text{in\_lane\_veh}[j] \implies \text{in\_lane\_same}[i,j] = 1 \\
\text{in\_lane\_same}[i,j] = 1 \implies \text{in\_in\_time}[j] \geq \text{in\_in\_time}[i] + \frac{\text{l\_veh}}{\text{V}};
\]

The second one is an indicator constraint, but you would just need to replace the first one by equivalent linear constraints.

Given that \(\text{in\_lan\_veh}\) can only be either 1 or 2, those constraints could be

\[
\text{in\_lane\_same}[i,j] \geq 3 - \text{in\_lane\_veh}[i] - \text{in\_lane\_veh}[j] \\
\text{in\_lane\_same}[i,j] \geq \text{in\_lane\_veh}[i] + \text{in\_lane\_veh}[j] - 3
\]
New Solver Interface Library (MP)

Design

- C++ library for building efficient, configurable solver drivers
- Support for features of current C interface library
- Extensive toolset for problem recognition and transformation

Motivation . . .

- AMPL has logical and “not linear” expressions for writing models the way you think of them
- Old interfaces have very limited support for these
- New interfaces, built with MP, allow these expressions to be used and combined freely
Example: Multi-Product Network Flow

Motivation
○ Ship products efficiently to meet demands

Context
○ a transportation network
  ■ Nodes representing cities
  ■ arcs representing roads
○ supplies at nodes
○ demands at nodes
○ capacities on arcs
○ shipping costs on arcs
Example: Multi-Product Network Flow

---

Decide

- how much of each product to ship on each arc

So that

- shipping costs are kept low
- shipments on each arc respect capacity of the arc
- supplies, demands, and shipments are in balance at each node
AMPL Model for Multi-Product Network Flow

---

set PRODUCTS;
set NODES;
param net_inflow {PRODUCTS,NODES};

set ARCS within {NODES,NODES};
param capacity {ARCS} >= 0;

param var_cost {PRODUCTS,ARCS} >= 0;
var Flow {PRODUCTS,ARCS} >= 0;

minimize TotalCost:
  sum {p in PRODUCTS, (i,j) in ARCS} var_cost[p,i,j] * Flow[p,i,j];

subject to Capacity {(i,j) in ARCS}:
  sum {p in PRODUCTS} Flow[p,i,j] <= capacity[i,j];

subject to Conservation {p in PRODUCTS, j in NODES}:
  sum {(i,j) in ARCS} Flow[p,i,j] + net_inflow[p,j] =
  sum {(j,i) in ARCS} Flow[p,j,i];
Example with conditions: Multi-Product Network Flow

---

Decide also

○ whether to use each arc

So that

○ variable costs plus fixed costs for shipping are kept low
○ shipments are not too small
○ not too many arcs are used
Positive Shipments Incur Fixed Costs

---

Linearization

```plaintext
param fix_cost {ARCS} >= 0;
var Use {ARCS} binary;

minimize TotalCost:
  sum {p in PRODUCTS, (i,j) in ARCS} var_cost[p,i,j] * Flow[p,i,j] +
  sum {(i,j) in ARCS} fix_cost[i,j] * Use[i,j];
```

How you think about it

```plaintext
param fix_cost {ARCS} >= 0;

minimize TotalCost:
  sum {p in PRODUCTS, (i,j) in ARCS} var_cost[p,i,j] * Flow[p,i,j] +
  sum {(i,j) in ARCS} if exists {p in PRODUCTS} Flow[p,i,j] > 0 then fix_cost[i,j];
```
Shipments Can’t Be Too Small

---

Linearization

subject to Min_Shipment \{(i,j) \text{ in ARCS}\}:
    \[\sum_{p \in \text{PRODUCTS}} \text{Flow}[p,i,j] \geq \text{min_ship} \times \text{Use}[i,j];\]

subject to Capacity \{(i,j) \text{ in ARCS}\}:
    \[\sum_{p \in \text{PRODUCTS}} \text{Flow}[p,i,j] \leq \text{capacity}[i,j] \times \text{Use}[i,j];\]

How you think about it

subject to Shipment_Limits \{(i,j) \text{ in ARCS}\}:
    \[\sum_{p \in \text{PRODUCTS}} \text{Flow}[p,i,j] = 0 \text{ or } \text{min_ship} \leq \sum_{p \in \text{PRODUCTS}} \text{Flow}[p,i,j] \leq \text{capacity}[i,j];\]
Can’t Use Too Many Arcs

---

Linearization

subject to Max_Used:
  sum {(i,j) in ARCS} Use[i,j] <= max_arcs;

How you think about it

subject to Limit_Used:
  atmost max_arcs {(i,j) in ARCS}
  (sum {p in PRODUCTS} Flow[p,i,j] > 0);
Linearization is Usually Not That Easy!

\[
\text{minimize total}_\text{fuelcost}: \\
\sum{(i,j) \in A} \sum{k \in V} X[i,j,k] \cdot \\
(\text{if } H[i,k] \leq 300 \text{ then } d\text{Mor}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 660 \text{ then } d\text{Aft}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 901 \text{ then } d\text{Eve}[i,j]) \cdot 5 + \\
(\text{if } H[i,k] \leq 300 \text{ then } t\text{Mor}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 660 \text{ then } t\text{Aft}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 901 \text{ then } t\text{Eve}[i,j]) \cdot 0.0504; \\
\]

subject to NoPersonIsolated
\[
\{l \in \text{TYPES['loc']}, r \in \text{TYPES['rank']}, j \in 1..\text{numberGrps}\}:
\sum {i \in \text{LOC_RANK}[l,r]} \text{Assign}[i,j] = 0 \text{ or} \\
\sum {i \in \text{LOC_RANK}[l,r]} \text{Assign}[i,j] + \sum {a \in \text{ADJACENT}[r]} \sum {i \in \text{LOC_RANK}[l,a]} \text{Assign}[i,j] \geq 2;
\]

subject to IfConstr \{i \in 1..\text{card(veh)}-1, j \in i+1..\text{card(veh)}:\}
\[
\text{veh}_\text{ind}[i] = \text{veh}_\text{ind}[j] \text{ and } \text{theory}_\text{time}[i] \leq \text{theory}_\text{time}[j] \\
\text{in}_\text{lane}_\text{veh}[i] = \text{in}_\text{lane}_\text{veh}[j] \\
\implies \text{in}_\text{in}_\text{time}[j] \geq \text{in}_\text{in}_\text{time}[i] + \text{l}_\text{veh}/V;
\]

minimize total_fuelcost:
\[
\sum{(i,j) \in A} \sum{k \in V} X[i,j,k] \cdot \\
(\text{if } H[i,k] \leq 300 \text{ then } d\text{Mor}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 660 \text{ then } d\text{Aft}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 901 \text{ then } d\text{Eve}[i,j]) \cdot 5 + \\
(\text{if } H[i,k] \leq 300 \text{ then } t\text{Mor}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 660 \text{ then } t\text{Aft}[i,j] \text{ else} \\
\text{if } H[i,k] \leq 901 \text{ then } t\text{Eve}[i,j]) \cdot 0.0504);
\]
Supported Extensions

Operators and functions

- Conditional: `if-then-else; ==>`, `<=`, `=<`
- Logical: `or`, `and`, `not`; `exists`, `forall`
- Piecewise linear: `abs`; `min`, `max`; `<breakpoints; slopes>`
- Counting: `count`; `atmost`, `atleast`, `exactly`; `numberof`
- Comparison: `>`, `<`, `!=`; `alldiff`
- Complementarity: `complements`
- Nonlinear: `*`, `/`, `^`; `exp`, `log`; `sin`, `cos`, `tan`; `sinh`, `cosh`, `tanh`
- Set membership: `in`

Expressions and constraints

- High-order polynomials
- Second-order and exponential cones (MOSEK driver!)
Supported Solvers

---

Solvers
- Gurobi, Xpress, COPT, MOSEK
- HiGHS, CBC, SCIP, GCG
- CPLEX soon

Modeling guide
- https://mp.ampl.com/model-guide.html

Examples using MP features
- https://colab.ampl.com
- https://ampl.com/mo-book
Quick introduction to amplpy!
What do you need to know to use amplpy?

- Basic Python features (lists, dictionaries, etc.)
- Data manipulation with Pandas dataframes
- How to model in AMPL (or how to ask Chat GPT to write AMPL models for you!)
Example: Christmas model (https://colab.ampl.com)

Christmas model by ChatGPT

Problem description, mathematical model, AMPL model and data copied directly from ChatGPT.
Example: N-Queens  (https://colab.ampl.com)

How can \( n \) queens be placed on an \( n \times n \) chessboard so that no two of them attack each other?

Constraint \texttt{alldiff} enforces a set of integer variables to take distinct values. Using \texttt{alldiff}, we can model N-Queens as follows:

\begin{verbatim}
param n integer > 0; # N-queens
var Row {1..n} integer >= 1 <= n;
s.t. row_attacks: alldiff ({j in 1..n} Row[j]);
s.t. diag_attacks: alldiff ({j in 1..n} Row[j]+j);
s.t. rdiag_attacks: alldiff ({j in 1..n} Row[j]-j);
\end{verbatim}
Example: N-Queens (https://colab.ampl.com)

N-Queens

Description: How can N queens be placed on an NxN chessboard so that no two of them attack each other?

Tags: amplpy, constraint-programming, highlights

Notebook author: Gleb Belov <gleb@ampl.com>

```
[1] # Install dependencies
!pip install --q amplpy

# Google Colab & Kaggle integration
from amplpy import AMPL, tools
ampl = tools.ampl_notebook(
    modules=['highs'],  # modules to install
    license_uid='default')  # license to use

Using default Community Edition License for Colab. Get yours at: https://ampl.com/ce
```
Example: Network design with redundancy ([https://colab.ampl.com](https://colab.ampl.com))
Wait a minute. How are AMPL & solvers running on Google Colab integrated with Python?
AMPL and all Solvers are now available as Python Packages

---

AMPL and all solvers are now available as python packages for **Windows**, **Linux** (**X86_64**, **aarch64**, **ppc64le**), and **macOS**.

```python
# Install Python API for AMPL
$ python -m pip install amplpy --upgrade

# Install solver modules (e.g., HiGHS, CBC, Gurobi)
$ python -m amplpy.modules install highs cbc gurobi

# Activate your license (e.g., free https://ampl.com/ce license)
$ python -m amplpy.modules activate <license-uuid>

# Import in Python
$ python
>>> from amplpy import AMPL
>>> ampl = AMPL() # instantiate AMPL object

> https://ampl.com/python/
```
AMPL is Free on Google Colab

---

> https://dev.ampl.com/ampl/python/colab.html

> https://try.ampl.com (quickly access to AMPL on Colab)

You can install AMPL on Google Colab (where it is free by default) as follows:

```
# Install dependencies
%pip install -q amplpy
```

```
# Google Colab & Kaggle integration
from amplpy import AMPL, tools
ampl = tools.ampl_notebook(
    modules=['gurobi', 'coin', 'highs', 'gokestrel'],  # modules to install
    license_uuid='default')  # license to use
```
Free licenses to use on Google Colab (and locally!)

- **ampl.com/ce**
  - For personal use
  - Immediate access without approvals required
  - No size-limits
  - Includes access to:
    - Open-source solvers
    - Commercial solver trials

- **ampl.com/courses**
  - For teaching
  - No size-limits
  - **Full access to all solvers!**
  - All students can use the license during the course.
The Python-first approach to learn and model with AMPL!
AMPL Model Colaboratory (https://colab.ampl.com)

Many examples: https://colab.ampl.com (live demo)
Data-Driven Mathematical Optimization with AMPL in Python


**MO-BOOK: Hands-On Optimization with AMPL in Python**

Welcome to this repository of notebooks Hands-On Optimization with AMPL in Python, also known as Data-Driven Mathematical Optimization with AMPL in Python, or MO-Book With AMPL, a project currently under development with completion expected by the end of 2023. This is the AMPL version of Hands-On Optimization in Python. These notebooks introduce the concepts and tools of mathematical optimization with examples from a range of disciplines. The goals of these notebooks are to:

- provide a foundation for hands-on learning of mathematical optimization,
- demonstrate the tools and concepts of optimization with practical examples,
- help readers to develop the practical skills needed to build models and solve problems using state-of-the-art modeling languages and solvers.

**Getting started**

The notebooks in this repository make extensive use of [AMPL](https://www.ampl.com) which is an interface that allows developers to access the features of AMPL from within Python. AMPL (A Mathematical Programming Language) is an algebraic modeling language to describe and solve high-complexity problems in large-scale optimization. Natural mathematical modeling syntax lets you formulate optimization models the
Deploying optimization applications quickly and easily using AMPL with Python

🐍
AMPL on Streamlit

> [https://ampl.com/streamlit](https://ampl.com/streamlit) (live demo)
Deploy anywhere with Docker

---

> https://dev.ampl.com/ampl/docker/

AMPL can be easily used on Docker containers and deployed anywhere.

```bash
# Use any image as base image with python installed
FROM python:3.9-slim-bullseye

# Install amplpy and all necessary amplpy.modules:
RUN python -m pip install amplpy --no-cache-dir # Install amplpy
RUN python -m amplpy.modules install highs gurobi --no-cache-dir # Install modules
```
Example project showing how to deploy applications

> https://amplpyfinance.ampl.com/

- How to use AMPL with Docker Containers:
  - A basic Docker Compose template for orchestrating a Flask application & a Celery queue with Redis.
  - https://github.com/ampl/amplpyfinance/tree/master/deployment/docker

- The same Docker images can be deployed to Kubernetes Clusters

- How to use AMPL in Continuous Integration Systems
  - This project uses Azure Pipelines and GitHub Actions for CI/CD
  - https://dev.ampl.com/ampl/cicd/
Continuous Integration Systems

- How to use AMPL in Continuous Integration Systems
  - This project uses **Azure Pipelines** and **GitHub Actions** for CI/CD
  - [https://dev.ampl.com/ampl/cicd/](https://dev.ampl.com/ampl/cicd/)

```
jobs:
  Test:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version: ["3.10"]
    steps:
      - uses: actions/checkout@v3
      - name: Set up Python
        uses: actions/setup-python@v4
        with:
          python-version: ${{ matrix.python-version }}
      - name: Install dependencies
        run: |
          set -ex
          python -m pip install -r requirements.txt
          python -m pip install amply
          python -m amply.modules install <solver1> <solver2>
          python -m amply.activate <license-uuid>
      - name: Install package
        run: |
          python -m pip install .
      - name: Test package
        run: |
          python -m <package-name>.tests
```
What about licenses for AMPL and Commercial Solvers?
Dynamic Licensing System

Leases Chart

Aggregate by:

Node ID
Free licenses to use on Google Colab (and locally!)

- ampl.com/ce
  - For personal use
  - Immediate access without approvals required!
  - No size-limits
  - Includes access to:
    - Open-source solvers
    - Commercial solver trials

- ampl.com/courses
  - For teaching
  - No size-limits
  - Full access to all solvers!
  - All students can use the license during the course.