Teaching, Learning, and Applying Optimization: New Developments in the AMPL Modeling System

Part II

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 121 A,
Technology Workshop, Saturday, October 14, 1:00 – 3:30 pm
Outline

— —

Part II (live demos):
- Quick introduction to amply (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
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](https://colab.ampl.com))

---

Christmas model by ChatGPT

Problem description, mathematical model, AMPL model and data copied directly from ChatGPT:

Certainly! Here is an example of a Christmas optimization model in AMPL:

```
# define the set of people
set PEOPLE;
# define the set of gifts
set GIFTS;
# define the cost of each gift
param cost(GIFTS); # define the happiness level for each person receiving each gift
param Happiness(PEOPLE, GIFTS);
# define the budget for gift-giving
param budget;
# define the decision variables:
var x[PEOPLE, GIFTS] binary;
# the objective is to max

minimize total_happiness:

sum {p in PEOPLE, g in GIFTS} Happiness(p, g) * x[p, g];
# subject to
subject to budget:
sum {g in GIFTS} cost[g] * x[*, g] <= budget;
```

Certainly! Here is the description of the Christmas optimization model, formatted in MathJax for use in a Colab notebook:

In this optimization problem, we are trying to choose the best gift for each person, given a set of people `PEOPLE` and a set of gifts `GIFTS`. In order to maximize the total happiness, each gift has a certain cost, and we have a budget that we cannot exceed. Each person can only receive one gift. The decision variables \( x[p, g] \) represent whether each person \( p \) receives gift \( g \), and must be binary (either 0 or 1).

The mathematical formulation of the problem is as follows:

\[
\begin{align*}
\text{minimize} & \quad \text{total_happiness} \\
\text{subject to} & \quad \text{budget} \\
\end{align*}
\]

Certainly! Here is an example of a Christmas optimization model in AMPL:

```
# define the set of people
set PEOPLE;
# define the set of gifts
set GIFTS;
# define the cost of each gift
param cost(GIFTS); # define the happiness level for each person receiving each gift
param Happiness(PEOPLE, GIFTS);
# define the budget for gift-giving
param budget;
# define the decision variables:
var x[PEOPLE, GIFTS] binary;
# the objective is to max

minimize total_happiness:

sum {p in PEOPLE, g in GIFTS} Happiness(p, g) * x[p, g];
# subject to
subject to budget:
sum {g in GIFTS} cost[g] * x[*, g] <= budget;
```

Certainly! Here is the description of the Christmas optimization model, formatted in MathJax for use in a Colab notebook:

In this optimization problem, we are trying to choose the best gift for each person, given a set of people `PEOPLE` and a set of gifts `GIFTS`. In order to maximize the total happiness, each gift has a certain cost, and we have a budget that we cannot exceed. Each person can only receive one gift. The decision variables \( x[p, g] \) represent whether each person \( p \) receives gift \( g \), and must be binary (either 0 or 1).

The mathematical formulation of the problem is as follows:

\[
\begin{align*}
\text{minimize} & \quad \text{total_happiness} \\
\text{subject to} & \quad \text{budget} \\
\end{align*}
\]
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 `alldiff` enforces a set of integer variables to take distinct values. Using `alldiff`, we can model N-Queens as follows:

```
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);
```

1 == 1

3+1 == 2+2

Row[1]-1 == Row[2]-2
1-1 == 2-2
Example: N-Queens ([https://colab.ampl.com](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>

```python
# Install dependencies
!pip install --quiet amplpy
```

```
# Google Colab & Kaggle integration
from amplpy import AMPL, tools
AMPL = toolsAMPLNotebook(
    modules=['higgs'], # modules to install
    license_uid='default') # license to use
```

Using default Community Edition License for Colab. Get yours at: [https://ampl.com/ce](https://ampl.com/ce)

Licensed to AMPL Community Edition License for the AMPL Model Colaboratory ([https://colab.ampl.com](https://colab.ampl.com)).
Example: Network design with redundancy (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.

# 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/
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

New Book:  https://ampl.com/mo-book (live demo)

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 amplpy 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/)

```yaml
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 amplpy
        python -m amplpy.modules install <solver1> <solver2>
        python -m amplpy.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.