Ayudantía 1 Python + Gurobi

MARIANA ORTEGA MVORTEGA2@UC.CL

Iterar en Python y sumas agregadas

- Los loops iteran sobre colecciones de elementos (listas, diccionarios,…)
  - Útil para representar el “para todo”
  - Ej: for c in cities:

        print(c) #recuerda indentar

- Listas por comprensión
  - Construir listas eficientemente via notación de conjuntos
  - Ej: obj = quicksum(cost[a] * x[a] for a in arcs)

- Combinar loops con sumas es ideal al construir restricciones!
  - Ej: $\sum_{j \in J} x_{i,j} \leq 5 \quad \forall i \in I$
  - for i in I:

        m.addConstr(quicksum(x[i,j] for j in J) $\leq$ 5)

  O… m.addConstrs(quicksum(x[i,j] <=5 for J in J) for i in I)

Parameter tuning tool

- Herramienta de ajuste de parámetros
- Mejorar tiempo de ejecución
- Automatiza la búsqueda de parámetros
- Archivo .mps(LP or MIP) or .lp

Para saber más de manejo de parámetros revisa:
http://www.gurobi.com/documentation/8.0/refman/parameters.html

Command Tune

```
C:\Users\Mariana>grbtune C:\Users\Mariana\Documents\UC\2017-2\Metodos\Proyecto\test.lp
```

1. Corrida de base (MIP: guarda tiempo de ejecución promedio)
2. Prueba distintas combinaciones de parámetros y las va imprimiendo

```
Testing candidate parameter set 3...

        NormAdjust 3
        Aggregate 0

Solving model ... runtime 0.15s

Improvement found:
  baseline: runtime 0.21s
  improved: runtime 0.15s
Total elapsed tuning time 1s (2s remaining)
```

3. Cuando la herramienta termina imprime un resumen con un detalle de los mejores sets de parámetros que encontró (TuneResults: N° sets) (default: tradeoff runtime y n° parámetros cambiados)

4. También, muestra los nombres de los archivos que creó con la información de ajuste

(Puedes ajustar TuneTimeLimit)

```
Tested 20 parameter sets in 97.89s

Baseline parameter set: runtime 3.38s

Improved parameter set 1 (runtime 1.62s):

        Method 2
        Heuristics 0
        VarBranch 1
        CutPasses 3
        GomoryPasses 0

Improved parameter set 2 (runtime 2.03s):

        Method 2
        Heuristics 0
        VarBranch 1
        CutPasses 3

Improved parameter set 3 (runtime 2.38s):

        Method 2
        VarBranch 1

Wrote parameter files tune1.prm through tune3.prm
Wrote log files: tune1.log through tune3.log
```

ComputeIIS

- Computa un Irreducible Inconsistent Subsystem (Subsistema inconsistente irreducible)
- Subsistema de restricciones y límites de variables con las siguientes propiedades:

1. El subsistema presentado por el IIS es infactible
2. Si cualquiera de las restricciones o límites se remueve, el subsistema se vuelve factible

- Un modelo infactible puede tener múltiples IIS
- OJO: Solo para quitar restricciones y límites de variables (NO para agregar)
- Ej:

$$model.computeIIS()$$
$$model.write("model.ilp")$$

VER ARCHIVO DE PROGRAMACIÓN

- “Modelo_infactible.ipynb” o “Modelo_infactible.py”
- Los archivos “AddVars.py” y “AddVar.py” son dos formas equivalentes de definir las variables, pero AddVars.py es más eficiente
- Los archivos “AddConstrs.py” y “AddConstr.py” son dos formas equivalentes de definir restricciones, pero AddConstrs.py es más eficiente (Evita problemas de memoria!!!!)
- “Planificación Empresa Equipo Rocket.py” es el archivo de la programación completa del problema de la fábrica del Equipo Rocket programado de forma eficiente

Ayudantía 1 Python + Gurobi

MVORTEGA2@UC.CL