



Prescriptive Analytics mit Python

Beleg Quadratic Assignment Problem

Quadratic Assignment Problem

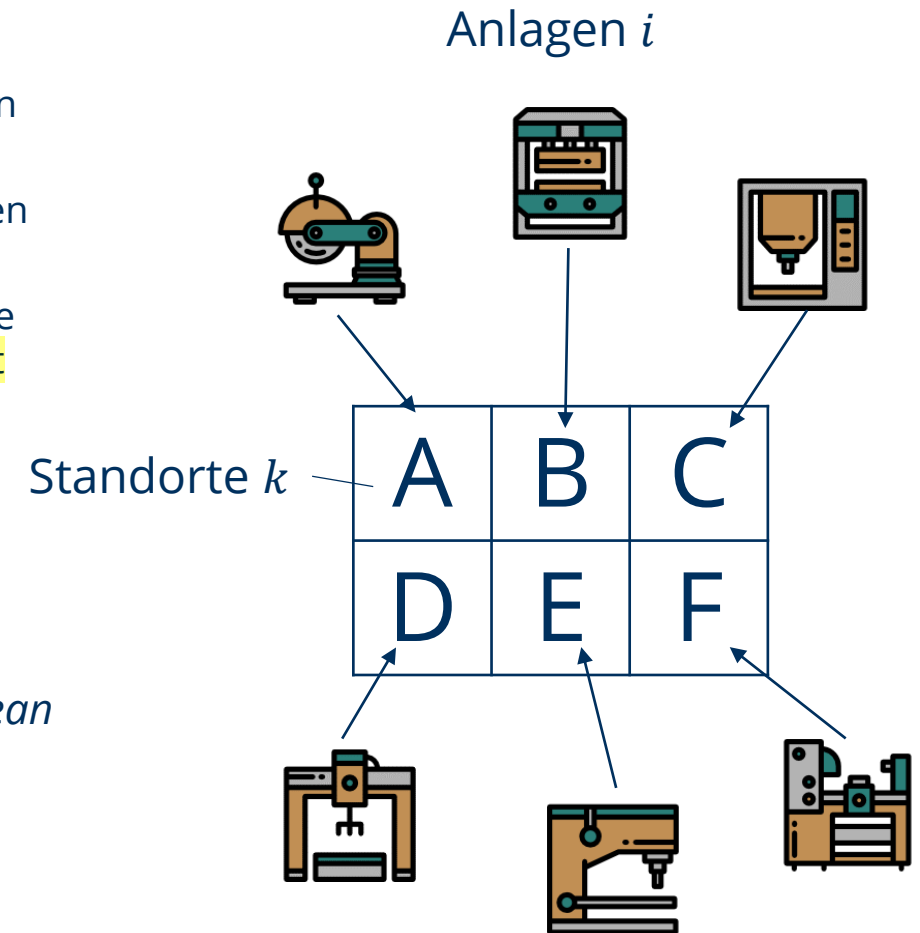
Planungsproblem

- n Anlagen müssen n Standorten in einer Fabrik zugeordnet werden
- Die Standorte werden vereinfacht als identisch und geeignet für alle Anlagen angenommen
- Zwischen den Anlagen sind entsprechend der geplanten Produktionsmengen die erforderlichen Transportmengen gegeben
- **Zielstellung** ist eine Zuordnung von Anlagen zu Standorten zu finden, welche die **Gesamtransportleistung** (transportierte Menge x Entfernung) minimiert
- Es existieren Inkompatibilitäten zwischen einzelnen Anlagen, d.h. für vorgegebene Anlagen muss ein Mindestabstand eingehalten werden

Startliteratur

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., & Querido, T. (2007). A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2), 657-690.

Burkard, R. E., Karisch, S. E., & Rendl, F. (1997). QAPLIB—a quadratic assignment problem library. *Journal of Global optimization*, 10(4), 391-403.



Erläuterungen zu Inputdaten

- Die Instanzen basieren auf der QAPLIB (<https://www.opt.math.tugraz.at/qaplib/>)
- Daten je Instanz:
 - Anzahl an Anlagen / Stationen = n
 - Matrix A : Anzahl der notwendigen Transporte a_{ij} von Anlage i zu Anlage j
 - Matrix D : Distanzen d_{kl} von Station k zu Station l [in m]
 - Matrix M : **Mindestabstände** m_{ij} zwischen Anlage i und Anlage j [in m]
- Testdatensätze: scr12.dat, scr15.dat, scr20.dat, tai25b.dat
- ACHTUNG: Die Matrizen der Instanzen müssen nicht symmetrisch sein
 - Es kann gelten $a_{ij} \neq a_{ji}$ oder $d_{kl} \neq d_{lk}$ oder $m_{ij} \neq m_{ji}$
- Für Problemstellung ohne Mindestabstände optimale Lösungen beigelegt

```
{
  "Size": 12,
  "A": [
    [0, 180, 120, 0, 0, 0, 0, 0, 0, 104, 112, 0],
    [180, 0, 96, 2445, 78, 0, 1395, 0, 120, 135, 0, 0],
    [120, 96, 0, 0, 0, 221, 0, 0, 315, 390, 0, 0],
    [0, 2445, 0, 0, 108, 570, 750, 0, 234, 0, 0, 140],
    [0, 78, 0, 108, 0, 0, 225, 135, 0, 156, 0, 0],
    [0, 0, 221, 570, 0, 0, 615, 0, 0, 0, 0, 45],
    [0, 1395, 0, 750, 225, 615, 0, 2400, 0, 187, 0, 0],
    [0, 0, 0, 0, 135, 0, 2400, 0, 0, 0, 0, 0],
    [0, 120, 315, 234, 0, 0, 0, 0, 0, 0, 0, 0],
    [104, 135, 390, 0, 156, 0, 187, 0, 0, 0, 36, 1200],
    [112, 0, 0, 0, 0, 0, 0, 0, 0, 36, 0, 225],
    [0, 0, 0, 140, 0, 45, 0, 0, 0, 1200, 225, 0]
  ],
  "D": [
    [0, 1, 2, 3, 1, 2, 3, 4, 2, 3, 4, 5],
    [1, 0, 1, 2, 2, 1, 2, 3, 3, 2, 3, 4],
    [2, 1, 0, 1, 3, 2, 1, 2, 4, 3, 2, 3],
    [3, 2, 1, 0, 4, 3, 2, 1, 5, 4, 3, 2],
    [1, 2, 3, 4, 0, 1, 2, 3, 1, 2, 3, 4],
    [2, 1, 2, 3, 1, 0, 1, 2, 2, 1, 2, 3],
    [3, 2, 1, 2, 2, 1, 0, 1, 3, 2, 1, 2],
    [4, 3, 2, 1, 3, 2, 1, 0, 4, 3, 2, 1],
    [2, 3, 4, 5, 1, 2, 3, 4, 0, 1, 2, 3],
    [3, 2, 3, 4, 2, 1, 2, 3, 1, 0, 1, 2],
    [4, 3, 2, 3, 3, 2, 1, 2, 2, 1, 0, 1],
    [5, 4, 3, 2, 4, 3, 2, 1, 3, 2, 1, 0]
  ]
}
```

- Zur Überprüfung der gefundenen Lösungen sollte ein einheitliches Output-Format verwendet werden
- Speicherung einer zulässigen Lösung im .json Format mit:
 - Zielfunktionswert (Gesamttransportleistung)
 - Permutation der den Stationen $1, \dots, n$ zugeordneten Anlagen

Beispiel:

```
{  
    "ObjVal": 1012,  
    "Permutation": [2, 9, 3, 1, 4, 7, 8, 10, 5, 6]  
}
```

- Für jede Instanz sind zwei Lösungen zu erzeugen: jeweils eine mit und ohne Berücksichtigung von Mindestabständen
- Die Lösung wird in einem separaten Ordner „Solutions“ abgespeichert.
- Der Dateiname der Lösung setzt sich wie folgt zusammen „<Name der Problem Instanz>_Lsg_<typ>“ z.B.:
scr12_Lsg_oMA.json (ohne Mindestabstände) oder **scr12_Lsg_mMA.json** (mit Mindestabständen)