Network Design

Part V TSP related problems

ATSP

Given

A directed graph $G=(N,A)$ and a cost $c_{ij} \geq 0$ for each edge in A

Find

The tour (a directed cycle that contains all *n* nodes) of minimal length

Example

For our discussion we work on complete graphs embedded in a planar grid

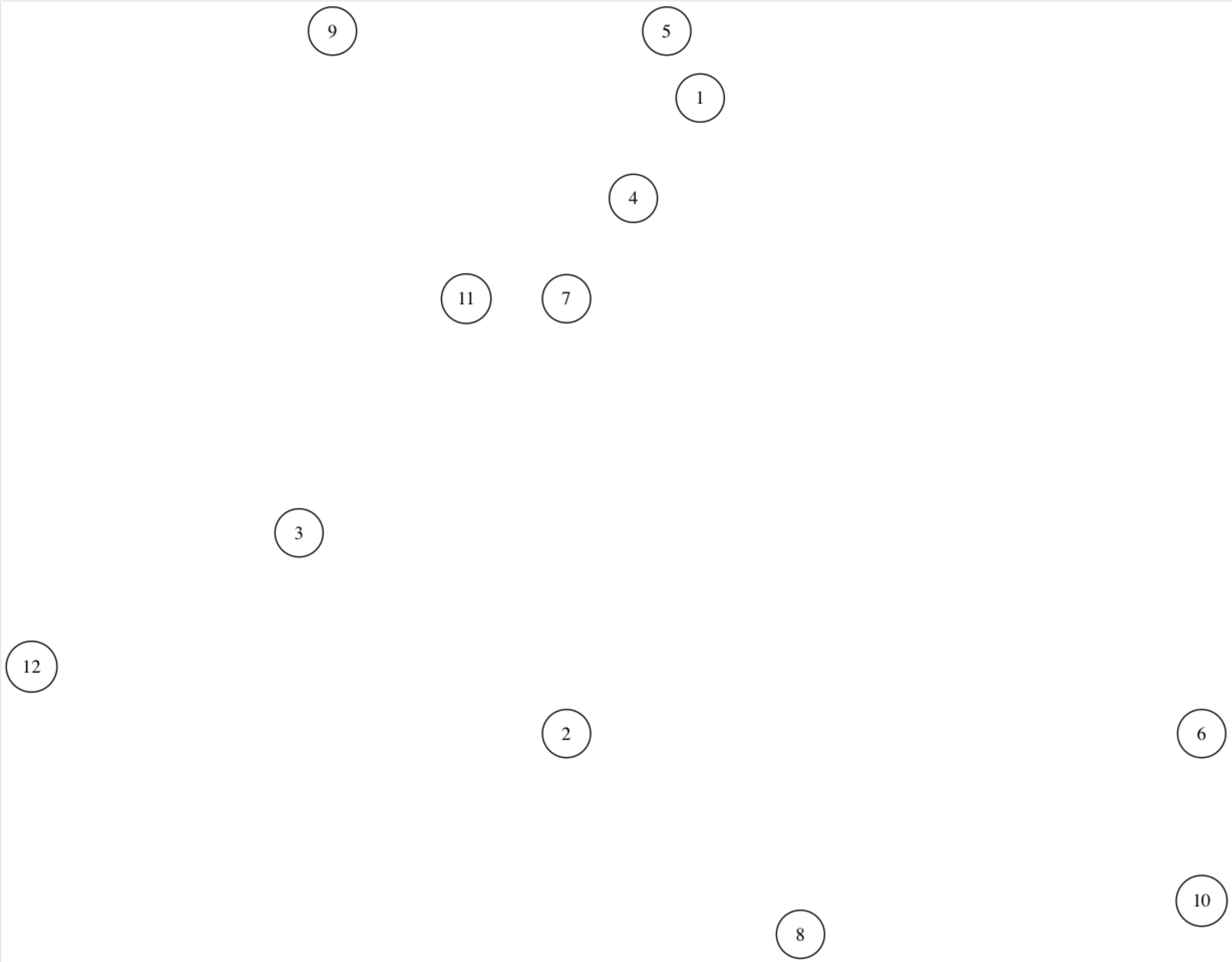
Distances between nodes are equal to the euclidean distances between nodes (scaled by an appropriate factor) plus a random perturbation

The code **generator_atsp.py** invoked as

```
generator_atsp.py 12 -x 40 -y 30 -p 100
```

returns a complete graph with 12 nodes embedded in a gride of size 40 x 30 cells

Example



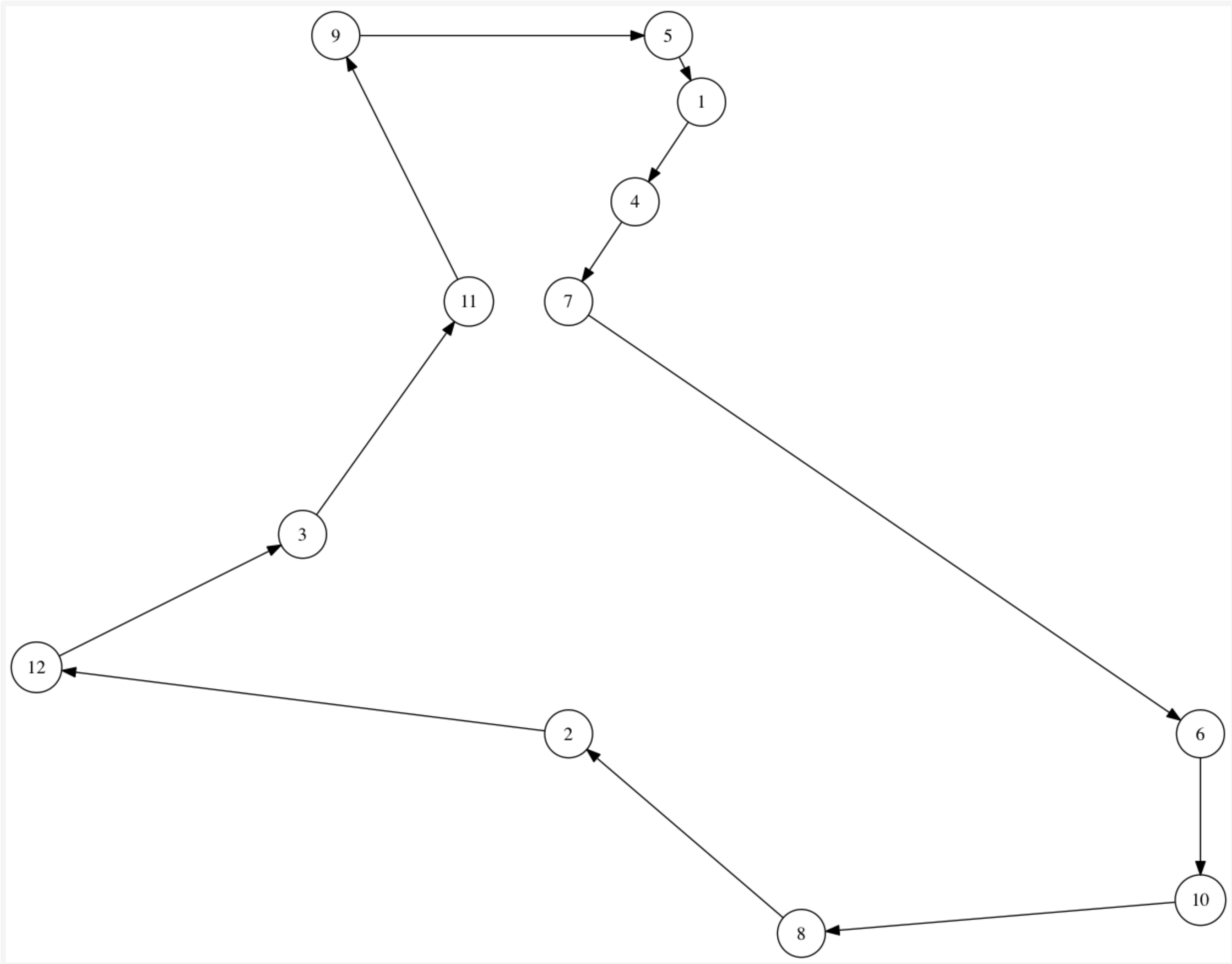
The graph is complete (arcs are not represented)

Example

Distance matrix

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0	2011	1795	380	287	2452	786	2607	1174	2919	1011	2653
2	1954	0	1053	1670	2208	1941	1370	925	2237	2055	1432	1628
3	1792	1008	0	1445	1937	2837	1157	1968	1562	2956	926	970
4	399	1644	1416	0	582	2426	379	2343	1051	2791	592	2295
5	272	2178	1914	534	0	2666	919	2790	1095	3114	1060	2742
6	2468	1926	2853	2356	2732	0	2349	1419	3432	536	2584	3539
7	817	1327	1111	362	906	2367	0	2060	1064	2685	322	1978
8	2519	975	1949	2292	2778	1359	2091	0	3059	1263	2170	2533
9	1118	2295	1577	1043	1085	3416	1067	3061	0	3770	979	2108
10	2850	1969	2965	2799	3117	537	2627	1226	3737	0	2890	3626
11	935	1395	897	603	1024	2655	371	2198	898	2890	0	1791
12	2644	1694	951	2323	2704	3537	1999	2476	2113	3608	1789	0

Example



Optimal solution: value 11623

Linear Programming formulation

Variables

$$x_{ij} = \begin{cases} 1 \text{ if arc } (i,j) \text{ is in the tour} \\ 0 \text{ otherwise} \end{cases}$$

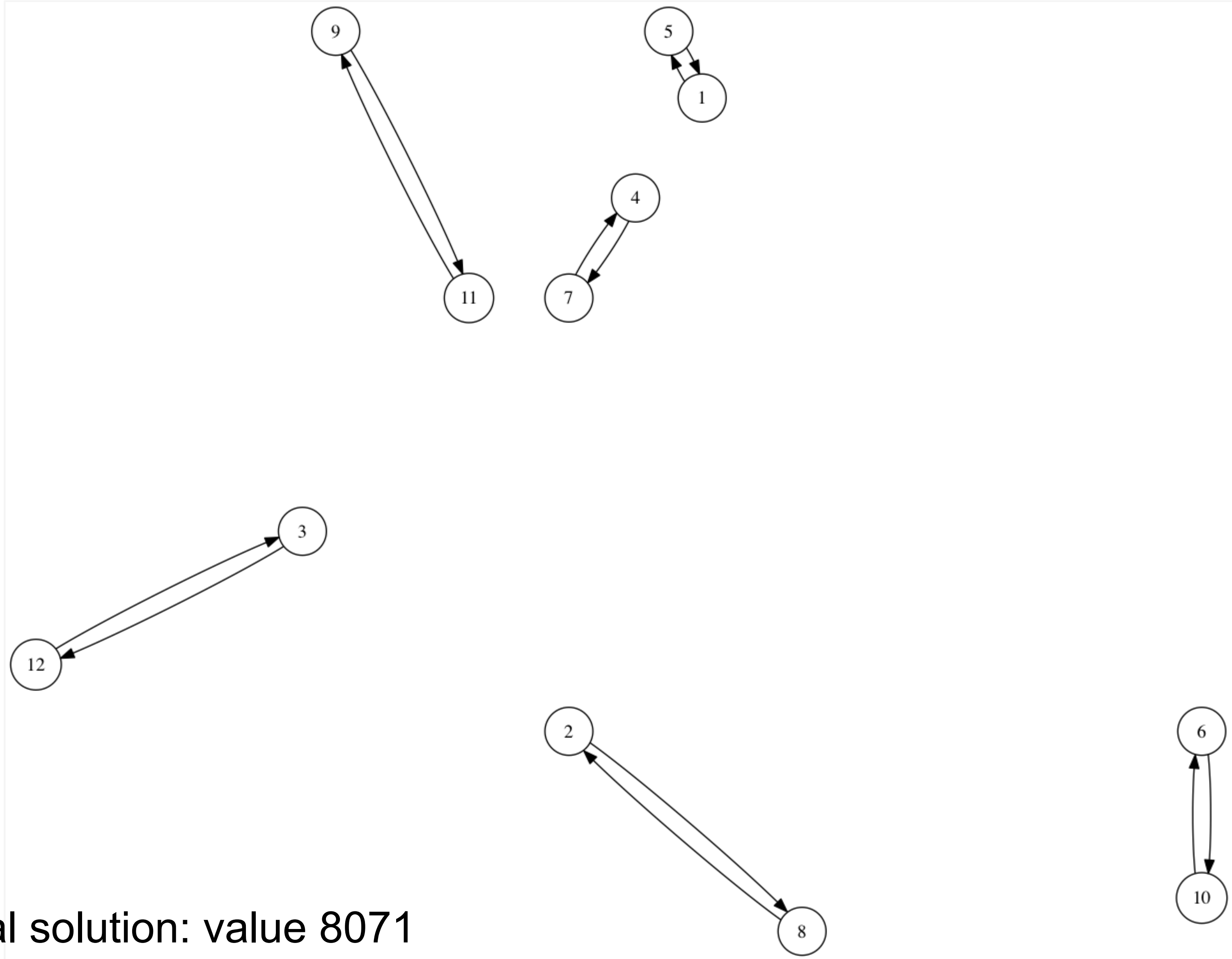
Assignment Constraints

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j\in\delta^+(i)} x_{ij} = 1 \;\; \forall i \in N \\
& \sum_{j\in\delta^-(i)} x_{ji} = 1 \;\; \forall i \in N \\
& 0 \le x_{ij} \le 1
\end{aligned}
$$

In the code

```
atsp = lp.LpProblem ("ATSP formulation", lp.LpMinimize)

x = lp.LpVariable.dicts ('x', G.edges(), lowBound=0, upBound=1,
cat=lp.LpContinuous)

# Objective function

atsp += lp.lpSum (G[i][j]['dist'] * x[(i,j)] for (i,j) in G.edges())

# Assignment contstraints

for i in G.nodes():
    name = "FS_",i
    atsp += lp.lpSum (x[(i,j)] for j in G.successors(i)) == 1, name

for i in G.nodes():
    name = "RS_",i
    atsp += lp.lpSum (x[(j,i)] for j in G.predecessors(i)) == 1, name
```

Example



Optimal solution: value 8071

First idea

Forbid all cycles of length 2

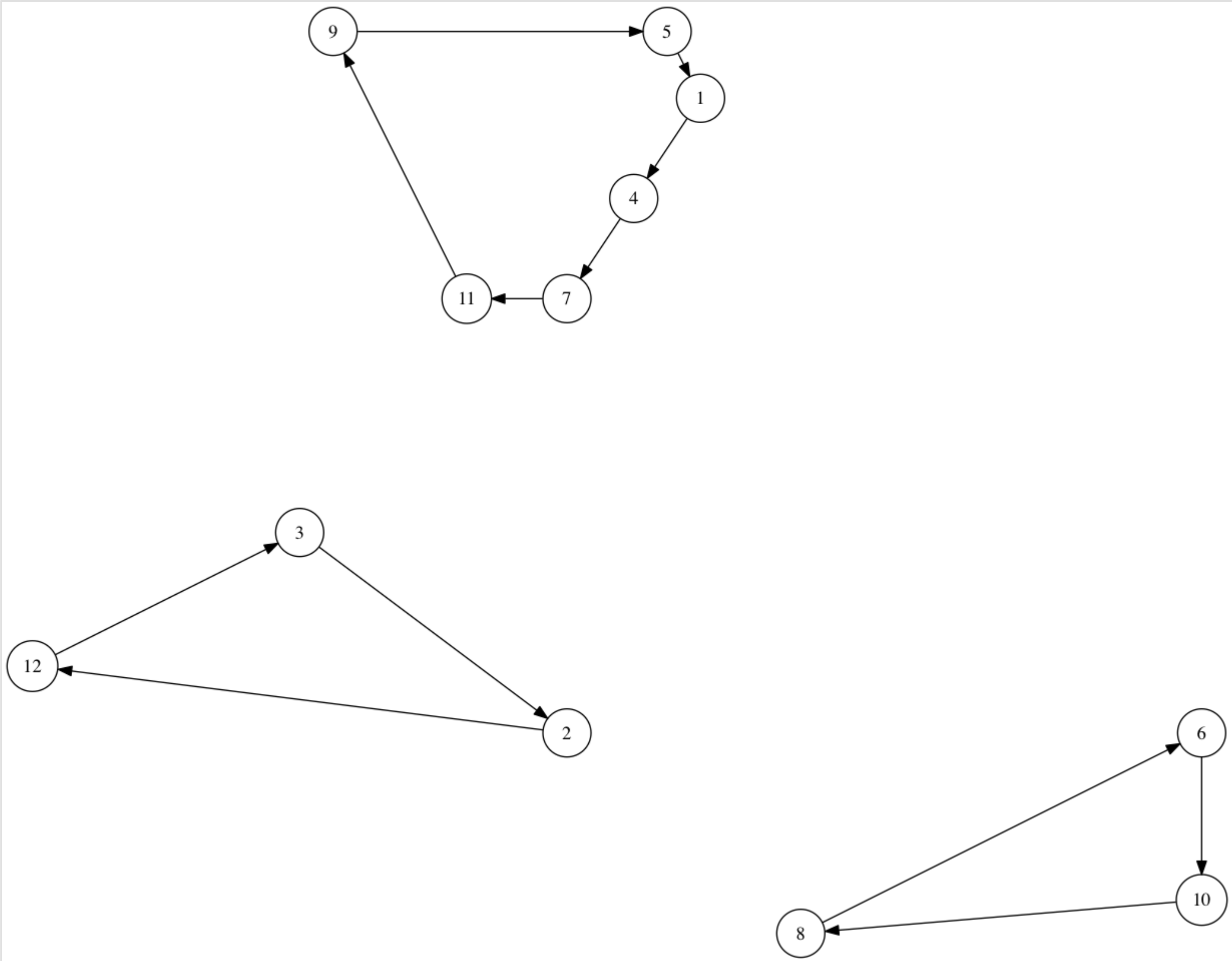
$$x_{ij} + x_{ji} \leq 1 \;\; \forall (i,j) \in A, i < j$$

Add the code:

```
for i in G.nodes():
    for j in G.nodes()[i:]:
          name="Sub_" + str(i) + "_" + str(j)
          atsp += x[(i,j)] + x[(j,i)] <= 1, name
```

and run again. You get the optimal solution:

Assignment constraints + 2-cycle elimination constraints



Optimal solution: value 10044

Miller Tucker Zemlin Subtour Elimination Constraints

To exclude all subtours, one can use extra variables u_i and the contraints:

$$u_i \in \mathbb{R}, \forall i \in n$$
$$u_1 = 1$$
$$2 \leq u_i \leq n \quad \forall i \neq n$$
$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}) \quad \forall i \neq 1, \forall j \neq 1.$$

The last set of inequalities are known as **arc-constraints**
This formulation is compact

Miller Tucker Zemlin Subtour Elimination Constraints

In the code:

```
for i in G.nodes()[1:]:
    for j in G.nodes()[1:]:
        if i != j:
            name ="MTZ" + str(i) +"_" + str(j)
            atsp +=  u[i] - u[j] +  (G.number_of_nodes() - 1) \
                  * x[(i,j)] <= G.number_of_nodes() - 2, name
```

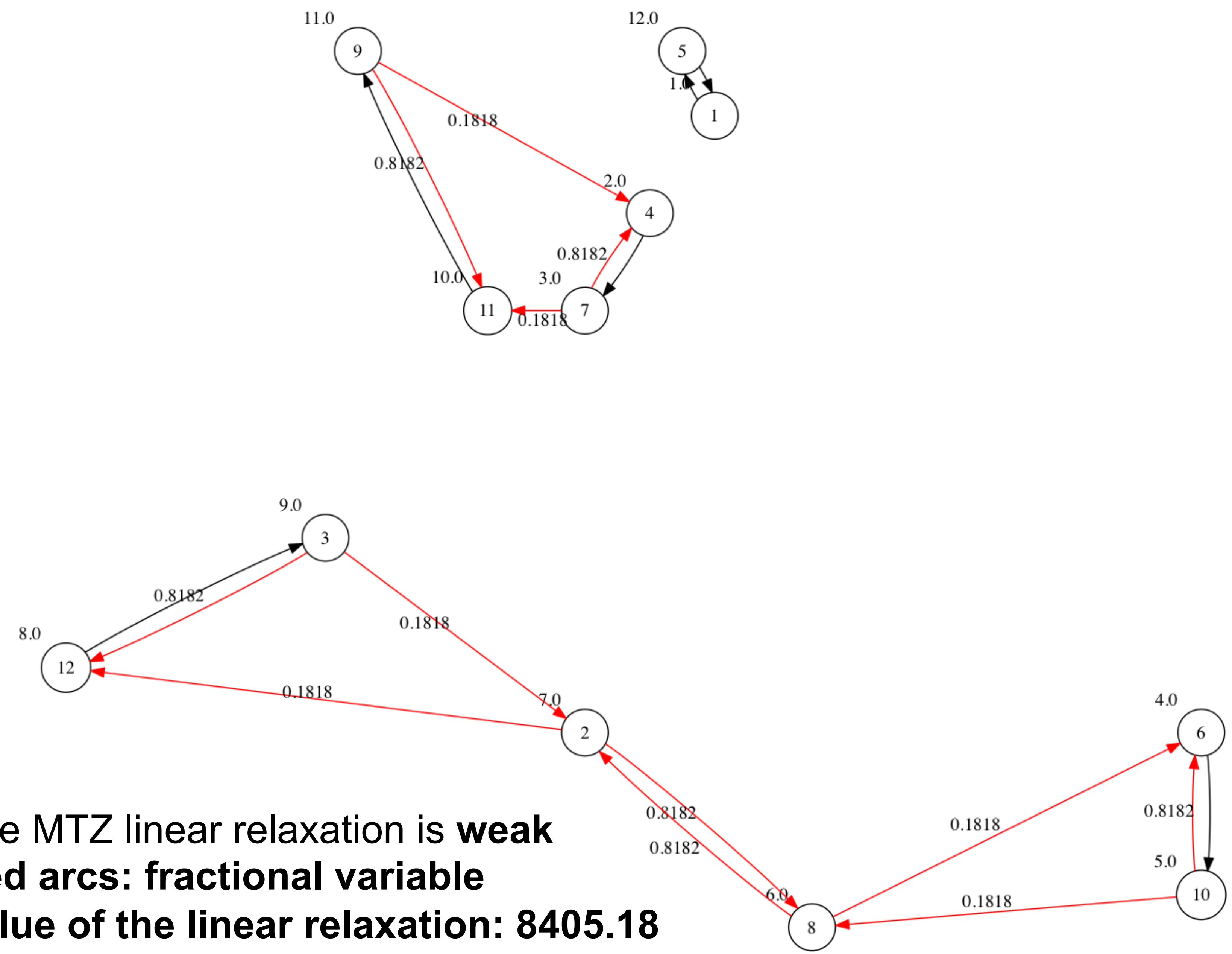
Remove the 2-cycle elimination constraints and change the *x* variable type:

```
    x = lp.LpVariable.dicts ('x', G.edges(), \
  lowBound=0, upBound=1, cat=lp.LpBinary)
```

Solver output

```
Solving LP relaxation...
GLPK Simplex Optimizer, v4.52
134 rows, 143 columns, 594 non-zeros
      0: obj =   2.259100000e+04  infeas =  6.875e+01 (1)
*    45: obj =   2.471363636e+04  infeas =  2.220e-16 (1)
*    97: obj =   8.405181818e+03  infeas =  5.433e-16 (1)
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
+    97: mip =     not found yet >=              -inf        (1; 0)
+   211: >>>>>   1.489400000e+04 >=   9.593000000e+03  35.6% (19; 0)
+   368: >>>>>   1.444400000e+04 >=   9.711000000e+03  32.8% (35; 7)
+   467: >>>>>   1.373100000e+04 >=   9.978000000e+03  27.3% (40; 14)
+   694: >>>>>   1.354300000e+04 >=   1.076000000e+04  20.5% (48; 38)
+   775: >>>>>   1.340900000e+04 >=   1.089100000e+04  18.8% (50; 51)
+   911: >>>>>   1.270600000e+04 >=   1.114900000e+04  12.3% (55; 65)
+   974: >>>>>   1.258900000e+04 >=   1.121300000e+04  10.9% (37; 123)
+  1042: >>>>>   1.194200000e+04 >=   1.128300000e+04   5.5% (45; 126)
+  1079: >>>>>   1.193000000e+04 >=   1.132100000e+04   5.1% (30; 161)
+  1255: >>>>>   1.185400000e+04 >=   1.142800000e+04   3.6% (30; 179)
+  1290: >>>>>   1.168000000e+04 >=   1.144500000e+04   2.0% (26; 188)
+  1318: >>>>>   1.162300000e+04 >=   1.148100000e+04   1.2% (17; 209)
+  1387: mip =   1.162300000e+04 >=     tree is empty   0.0% (0; 261)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   0.1 secs
Memory used: 0.4 Mb (412725 bytes)
```

Linear relaxation



The MTZ linear relaxation is **weak**

Red arcs: fractional variable

Value of the linear relaxation: 8405.18

Lifted MTZ Subtour Elimination Constraints

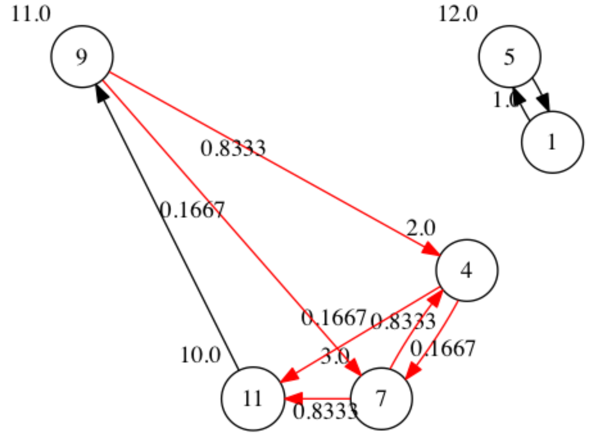
Desrochers and Laporte (1991) proposed the following strengthening of MTZ arc constraints

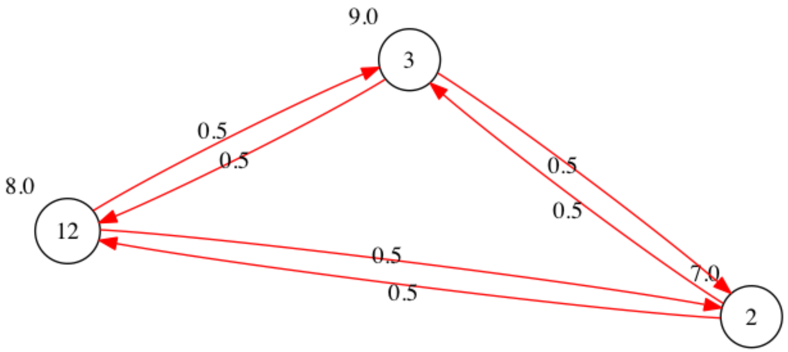
$$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \leq n-2$$
$$\forall i \neq j = 2, \ldots, n$$

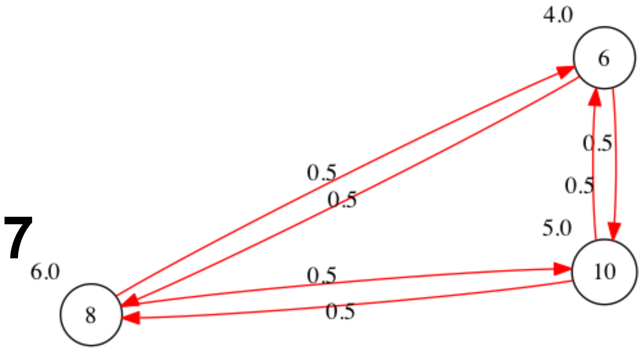
Code

```
for i in G.nodes()[1:]:
    for j in G.nodes()[1:]:
        if i != j:
            name ="MTZ_lifted" + str(i) + "_" + str(j)
            atsp +=  u[i] - u[j] + (G.number_of_nodes() - 1) * x[(i,j)]\
+ (G.number_of_nodes() - 3) * x[(j,i)]<= G.number_of_nodes() - 2, name
```

Linear relaxation







Value of the linear relaxation: 10069.17
Red arcs: fractional variables

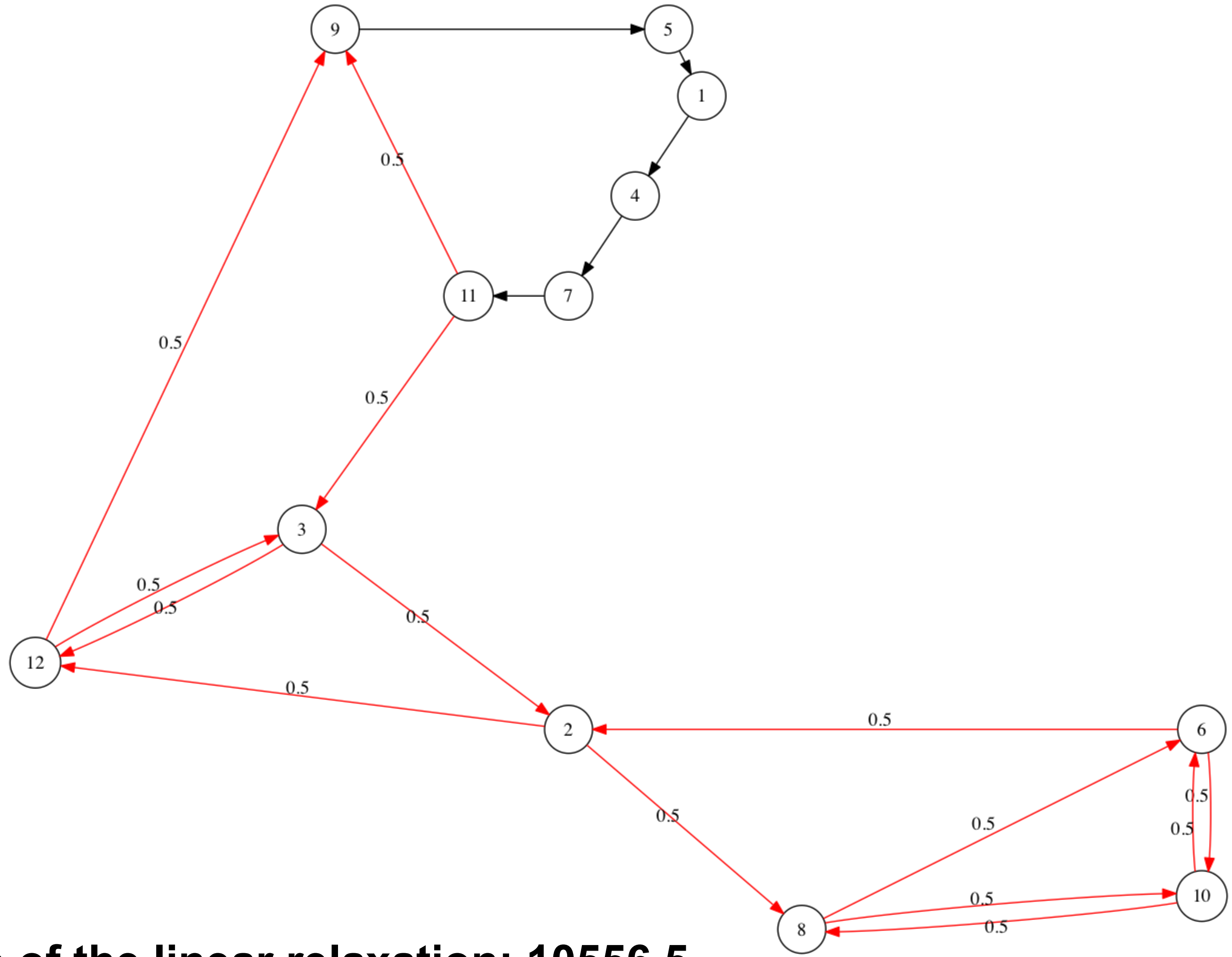
Dantzig Fulkerson Johnson Subtour Elimination Constraints

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \ \ \forall S \subsetneq N, |S| \geq 2$$

These constraints are **exponential in number** but are known to be facet defining for the ATSP polytope under mild conditions.

SECs separation algorithm: see MST slides

Linear relaxation



Value of the linear relaxation: 10556.5
Red arcs: fractional variables

Observations

The **largest lower bound** has been obtained by the DFJ constraints

However, embedding the constraints generation algorithm into the enumeration scheme requires particular care.

Such an approach is called **branch-and-cut.**

A simpler way to combine the strength of the DFJ constraints and the compactness of the lifted-MTZ formulation consists in merging the two formulations.

A possible approach

1. Start with the lifted MTZ formulation plus the 2-cycle elimination constraints
2. Run the DFJ separation algorithm before beginning the enumeration
3. Run the branch-and-bound

Important note

TSP exact approaches are very sophisticated (see CONCORDE) but the previous scheme can be useful in solving small TSP-derived problems

Solver log

Value of the initial relaxation [lifted-MTZ + 2-cycle elimination]: 10158

Add the violated subtour inequality
S = [12, 3, 2]

Improved lower bound: 10556.5

Branch-and-bound

Integer optimization begins...

```
+   133: mip =     not found yet >=              -inf        (1; 0)
+   193: >>>>>   1.276200000e+04 >=   1.106900000e+04  13.3% (10; 0)
+   340: >>>>>   1.175700000e+04 >=   1.140400000e+04   3.0% (22; 8)
+   376: >>>>>   1.162300000e+04 >=   1.155200000e+04   0.6% (13; 23)
+   409: mip =   1.162300000e+04 >=     tree is empty   0.0% (0; 59)
```

Further notes

The code atsp.py contains 2 copies of the problem.

The pulp object atsp contains only continuous variables
The pulp object atsp_int cpntains binary variables

Both objects contains the same set of contraints.

You can comment/uncomment part of the code to replicate the exercise discussed in this material

Drawing functions

`DrawSol (x)` draws the current solution (integer variables are represented by black arcs, red arcs correspond to fractional variables)
`DrawSolMTZ (x,u)` draws the values of the u variables associated to each node
`DrawSubtour (x, subtour)` draws the violated subtour found during the cutting plane

mTSP

Given

A directed graph $G=(N,A)$ and a cost $c_{ij} \geq 0$ for each edge in A

Find

A set of routes for *m* salesmen who all start from and turn back to a home city (depot)

Linear Programming formulation

Variables

$$x_{ij} = \begin{cases} 1 \text{ if arc } (i,j) \text{ is in the tour} \\ 0 \text{ otherwise} \end{cases}$$

LP formulation

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=2}^{n} x_{1j} = m \\
& \sum_{i=2}^{n} x_{i1} = m \\
& \sum_{j\in\delta^{+}(i)} x_{ij} = 1 \quad i = 2, \ldots, N \\
& \sum_{j\in\delta^{-}(i)} x_{ji} = 1 \quad i = 2, \ldots, N \\
& \text{Subtour Elimination Constraints} \\
& 0 \leq x_{ij} \leq 1
\end{aligned}
$$

Subtour Elimination Constraints

DFJ constraints

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \ \ \forall S \subseteq N \setminus \{1\}, S \neq 0$$

MTZ constraints

$$u_i - u_j + p x_{ij} \leq p - 1 \ \text{ for } 2 \leq i \neq j \leq n$$

+ Capacitated Vehicle Routing Problem

Given

A fleet of identical vehicles, with limited capacity Q located at a depot {0}.
A set of n customers with demand q_i>0.
A complete directed graph $G=(N,A)$ with $N=\{0,1,..,n\}$ and a traveling cost $c_{ij} \geq 0$ for each edge in A

Find

A minimum-cost collection of vehicle routes, each starting and ending at the depot, such that each customer is visited by exactly one vehicle, and no vehicle visits a set of customers whose total demand exceeds Q.

CVRP: 2 index formulation

$$x_{ij} = \begin{cases} 1 \text{ if some vehicle travels from } i \text{ to } j, \\ 0 \text{ otherwise.} \end{cases}$$

$$q(S) = \sum_{i \in S} q_i, \text{ for any } S \subset N_c$$

CVRP: 2 index formulation

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{s.t.} \sum_{i\in N_c} x_{ij} = 1 \;\forall j \in N_c$$

$$\sum_{j\in N_c} x_{ij} = 1 \;\forall i \in N_c$$

$$x(\delta^+(S)) \geq \lceil q(S)/Q \rceil \;\forall S \subseteq N_c$$

$$x \in \{0,1\}^{|A|}$$

CVRP: 2 index formulation single commodity

$$f_{ij} = \{\text{Load carried from } i \text{ to } j\ \}$$

$$\sum_{j \in \delta^-(i)} f_{ji} - \sum_{j \in \delta^+(i)} f_{ij} = q_i \ \forall i \in N_c$$

$$0 \leq f_{ij} \leq Q x_{ij}$$