MEMOCO — 15/11/2024

$\boxed{\text{TSP}}$ Company produces boards
- CPLEX
- DOCPLEX

BOARDS WITH HOLES $\rightarrow \underbrace{\sum \text{min TIME}}$

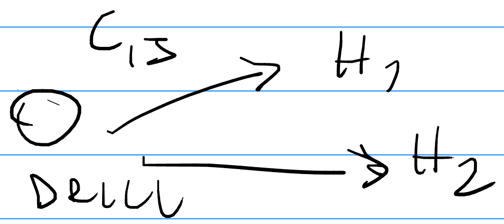
OBJECTIVE FUNCTION $\rightarrow [z]$

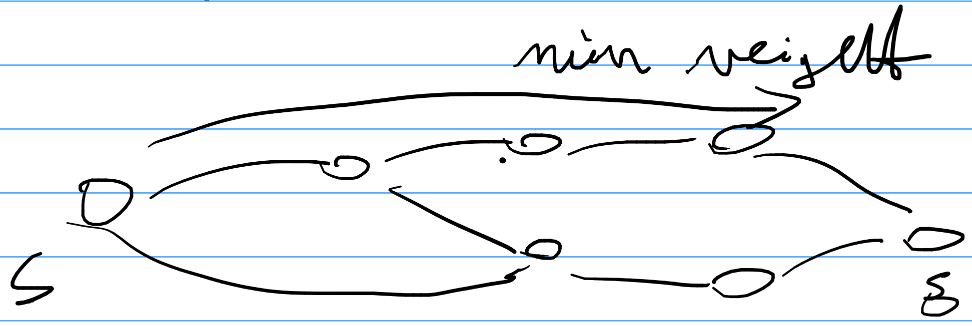
TSP $\rightarrow G = (N, A)$

N = set of nodes $\quad \forall\, i, j \in N$

A = set of arcs

c_{ij} = WEIGHT





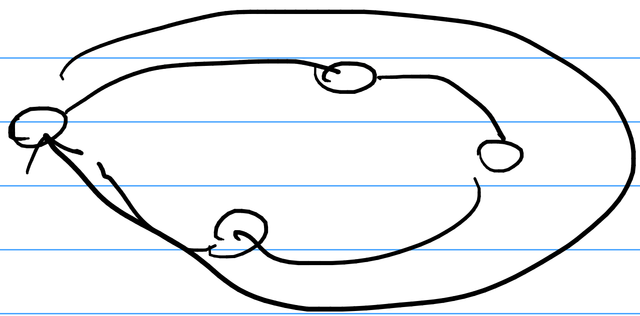
START - END

TSP with HAM CYCLES

→ cycle visiting each vertex once

SOLUTION → Use a COMPACT FORMULA

TSP → $G = (N, A)$



Sets:

- N = graph nodes, representing the holes;
- A = arcs (i, j), $\forall\ i, j \in N$, representing the trajectory covered by the drill to move from hole i to hole j.

Parameters:

- c_{ij} = time taken by the drill to move from i to j, $\forall\ (i, j) \in A$;
- 0 = arbitrarily selected starting node, $0 \in N$.

Decision variables:

- x_{ij}= amount of the flow shipped from i to j, $\forall\ (i,j) \in A$;
- y_{ij}= 1 if arc (i,j) ships some flow, 0 otherwise, $\forall\ (i,j) \in A$.

Integer Linear Programming model:

$$
\begin{array}{llllll}
\min & \displaystyle\sum_{i,j:(i,j)\in A} c_{ij} y_{ij} & & & & (1) \\
s.t. & \displaystyle\sum_{j:(0,j)\in A} x_{0j} & = & |N|-1 & & (2) \\
& \displaystyle\sum_{i:(i,k)\in A} x_{ik} - \sum_{j:(k,j)} x_{kj} & = & 1 & \forall\ k \in N \setminus \{0\} & (3) \\
& \displaystyle\sum_{j:(i,j)\in A} y_{ij} & = & 1 & \forall\ i \in N & (4) \\
& \displaystyle\sum_{i:(i,j)\in A} y_{ij} & = & 1 & \forall\ j \in N & (5) \\
& x_{ij} & \leq & (|N|-1)\ y_{ij} & \forall\ (i,j) \in A & (6) \\
& x_{ij} \in \mathbb{R}_+ & & & \forall\ (i,j) \in A & (7) \\
& y_{ij} \in \{0,1\} & & & \forall\ (i,j) \in A & (8)
\end{array}
$$

→ COSTO MINIMO

FLOW = N. NODI - 1 — INIZIO

→ BILANCIAMENTO FLUSSO

INIZIAMO DA N0 E CI ASPETTIAMO

(4) (5)

Each hole must have exactly one hole drilled before it

Each hole must have exactly one hole drilled after it

These create the Hamiltonian cycle (visit each node exactly once)

(6)

Forces yij = 1 if there is any flow on arc (i,j)

Links flow variables to path variables

|N|-1 is the maximum possible flow on any arc

This can be eliminated because:

- By constraint (3)/now (10), each non-starting node must receive 1 unit more flow than it sends
- Since there are |N|-1 nodes besides the starting node, this automatically forces |N|-1 units to flow from the starting node
- By optimality, we won't send more than necessary since this would only increase cost

2. Eliminating Flow to Starting Node: We can fix xi0 = 0 (flows to node 0) because:

- Some node must be visited last and connect back to 0 (handled by y variables)
- But there's no need to send flow back to 0 since it doesn't need to receive a unit of flow

3. The Simplified Model's Key Changes:

- Objective (9) remains the same: minimize total travel cost
- Constraint (10) now only considers flow balance for non-starting nodes and ignores flow back to 0
- Constraints (11) and (12) ensure each node is visited exactly once (Hamiltonian cycle)
- Constraint (13) links flows to path variables but only for j≠0
- Variables x and y are defined only where needed (14-15)

REST OF WIGIONS

[GAVISH-GRAVES APPROACH]

(1978)

WEAKER LP RELAXATION ?!

(DRAWBACK) ...

A "weaker LP relaxation" refers to how well the linear programming relaxation (where integer/binary constraints are replaced with continuous bounds) approximates the integer solution space.

Let me explain with an example:

Consider two different formulations F1 and F2 of the same TSP:

LP Relaxation Value vs Integer Optimal:

Copy

```
F1: LP relaxation optimal value = 80
F2: LP relaxation optimal value = 95
True integer optimal value = 100
```

Here, F2 has a "stronger" LP relaxation because:

- It provides a tighter (closer) bound to the true integer optimal value
- 95 is closer to 100 than 80 is
- This means F2's feasible region better approximates the true integer solution space

ALTERNATIVELY:
(FIRST, IMPLEMENT GAVISH/GRAVES ...)

⇒

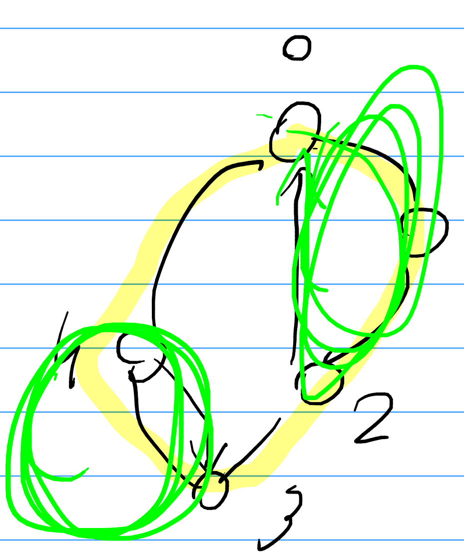
Other Important TSP Formulations:

Miller-Tucker-Zemlin (MTZ) formulation

Dantzig-Fulkerson-Johnson formulation

Christofides algorithm for metric TSP

Modern cutting plane approaches



PROBLEM: SUBTOURS!

(MEANS WE MIGHT COME BACK TO AN ALREADY DRIVED HOLE!)

IF THE FLOW DOES NOT FORWARD (MIGHT STALL!)

Example of Subtours

Suppose the graph has 5 nodes: $N = \{0, 1, 2, 3, 4\}$. A valid TSP solution would visit all nodes in a single cycle, such as:

$$0 \to 1 \to 2 \to 3 \to 4 \to 0$$

However, a solution containing **subtours** might look like this:

- $0 \to 1 \to 2 \to 0$ (subtour 1),
- $3 \to 4 \to 3$ (subtour 2).

Here, the salesman completes two smaller cycles but fails to connect all nodes in one complete tour.

Comparison with the Network Flow Model

Aspect	Network Flow Model	Gavish and Graves' Approach
Formulation Type	Compact ILP with flow variables.	Lagrangian relaxation + decomposition.
Strength	Polynomial-sized constraints, intuitive flow interpretation.	Dynamically handles subtours via decomposition.
Weakness	Requires explicit constraints for subtours.	Iterative subtour elimination; needs good initialization.
Scalability	Limited due to the quadratic number of variables.	Limited computational tests to 42 nodes.
Extensions	Primarily TSP and small variants.	Broader: Multi-TSP, Delivery, Dial-a-Bus, etc.
Implementation	Direct ILP solvers (Cplex, Gurobi, etc.).	More complex due to relaxation and tuning.