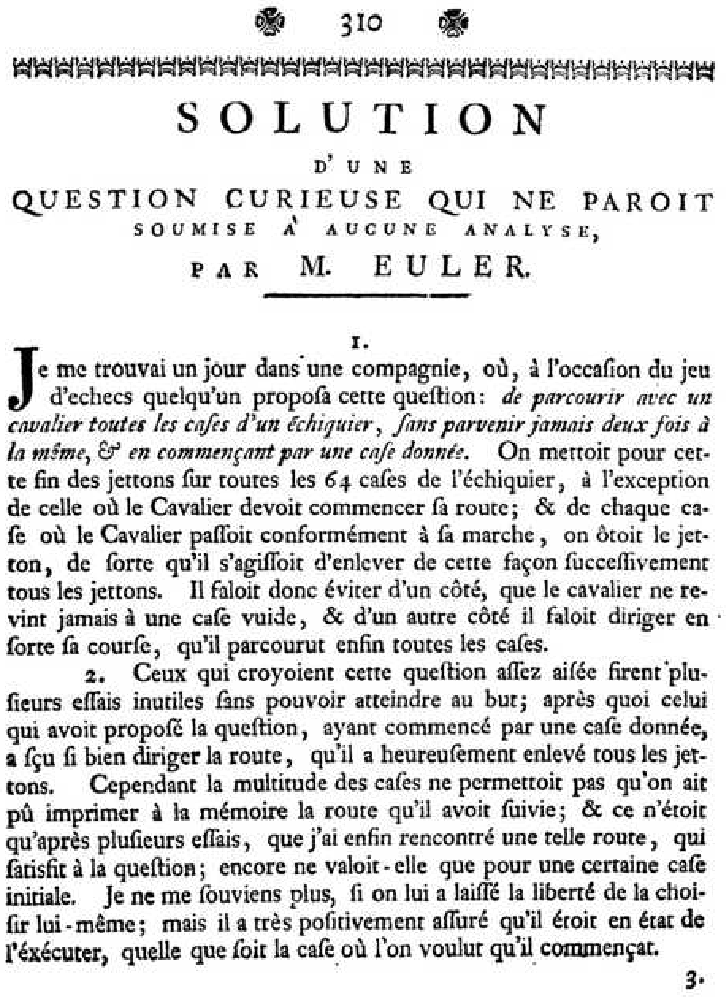
Operations Research (Master's Degree Course)

7.4 Problems on Graphs: Optimal Circuits

Silvano Martello

DEI "Guglielmo Marconi", Università di Bologna, Italy

The origins: circuits on special graphs



310

SOLUTION

D'UNE

QUESTION CURIEUSE QUI NE PAROIT SOUMISE A' AUCUNE ANALYSE,

PAR M. EULER.

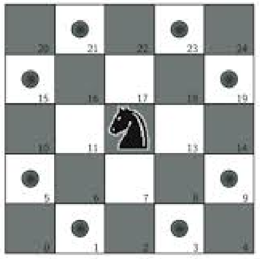
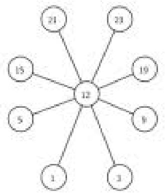
1.

Je me trouvai un jour dans une compagnie, où, à l'occaſion du jeu d'echecs quelqu'un propoſa cette queſtion: *de parcourir avec un cavalier toutes les caſes d'un échiquier, ſans parvenir jamais deux fois à la même, & en commençant par une caſe donnée.* On mettoit pour cette fin des jettons ſur toutes les 64 caſes de l'échiquier, à l'exception de celle où le Cavalier devoit commencer ſa route; & de chaque caſe où le Cavalier paſſoit conformément à ſa marche, on ôtoit le jetton, de ſorte qu'il s'agiſſoit d'enlever de cette façon ſucceſſivement tous les jettons. Il faloit donc éviter d'un côté, que le cavalier ne revint jamais à une caſe vuide, & d'un autre côté il faloit diriger en ſorte ſa courſe, qu'il parcourut enfin toutes les caſes.

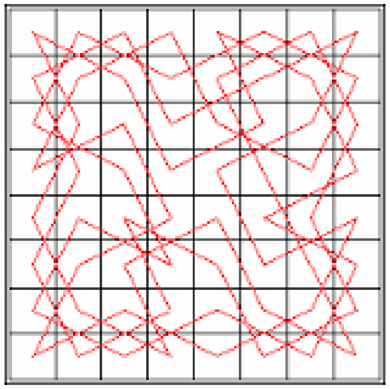
2. Ceux qui croyoient cette queſtion aſſez aiſée firent pluſieurs eſſais inutiles ſans pouvoir atteindre au but; après quoi celui qui avoit propoſé la queſtion, ayant commencé par une caſe donnée, a ſçu ſi bien diriger la route, qu'il a heureuſement enlevé tous les jettons. Cependant la multitude des caſes ne permettoit pas qu'on ait pû imprimer à la mémoire la route qu'il avoit ſuivie; & ce n'étoit qu'après pluſieurs eſſais, que j'ai enfin rencontré une telle route, qui ſatisfit à la queſtion; encore ne valoit-elle que pour une certaine caſe initiale. Je ne me ſouviens plus, ſi on lui a laiſſé la liberté de la choiſir lui-même; mais il a très poſitivement aſſuré qu'il étoit en état de l'éxécuter, quelle que ſoit la caſe où l'on voulut qu'il commençat.

3.

- **Leonard Euler** (1759): *Solution d'une question curieuse qui ne paroit soumise à aucune analyse*. **Knight's tour problem**:
  - a knight is placed on an empty $n \times n$ chessboard and must visit each square exactly once by only using valid chess moves of a knight:

  - by defining a graph in which the vertices correspond to the chessboard squares and the edges to the legal knight moves, a knight's tour corresponds to a path (or a cycle) that visits every vertex of the graph exactly once:

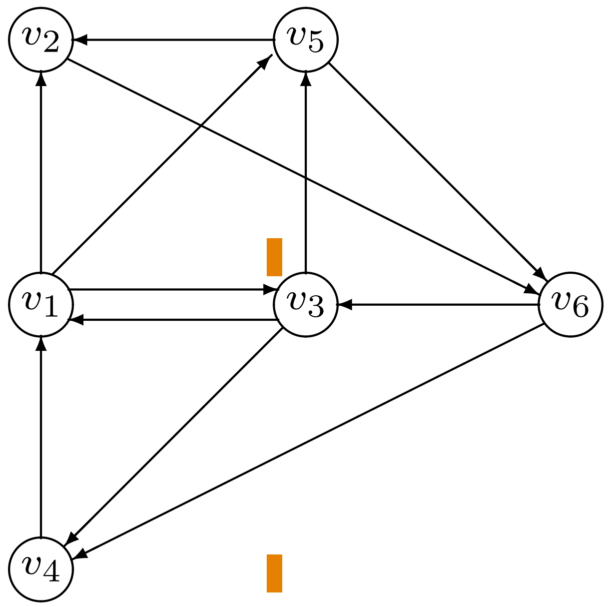


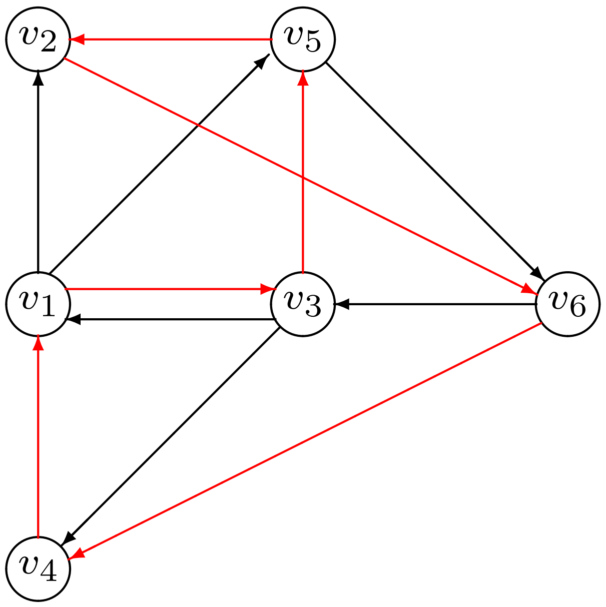
- Sir William Rowan Hamilton, Irish mathematician (1859): **Icosian Game**
  - played on a wooden planar representation of the edges of a dodecahedron (a graph), with holes at each of the twenty vertices:

  - The first player stuck five pegs in any consecutive vertices, and the second player was requested to stick the remaining fifteen pegs so as to complete the resulting path to a cycle visiting each vertex exactly once;
  - a circuit that passes through each vertex exactly once is called a **Hamiltonian circuit**;
  - sold for £25 to a Dublin toy manufacturer. (It seems that the sales were not satisfactory though.)

Hamiltonian circuits

Given a directed graph $G = (V, A)$, $V = \{v_1, \dots, v_n\}$, $A = \{(v_i, v_j) : v_i, v_j \in V\}$
or an undirected graph $G = (V, E)$, $E = \{(v_i, v_j) \equiv (v_j, v_i) : v_i, v_j \in V\}$
decide if the graph possesses a Hamiltonian circuit:

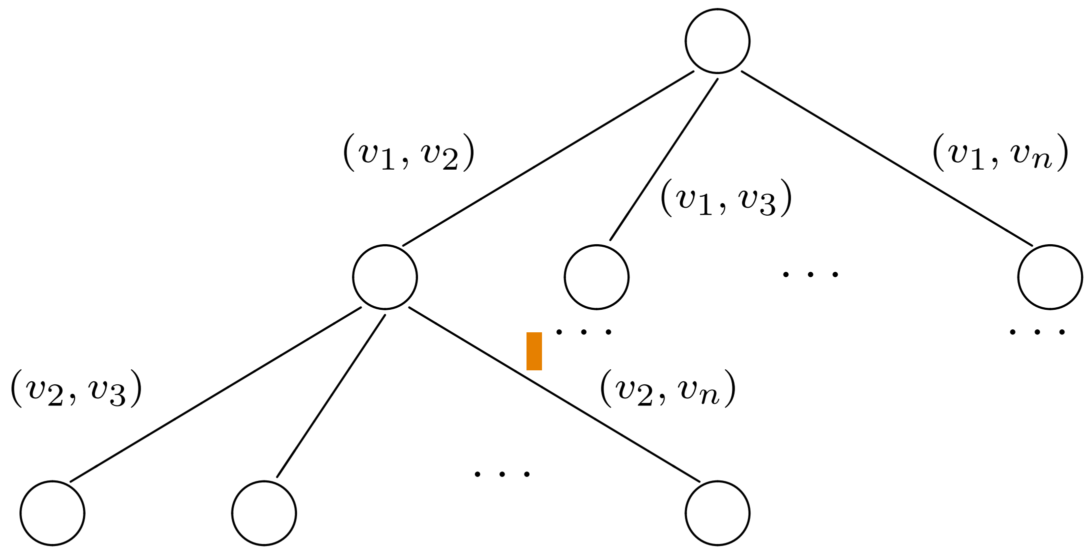




The problem can be solved by enumerating all permutations of the vertices and checking each of the for feasibility: This will take a time proportional to $(n-1)!$ ($\underline{\text{exponential}}$, impractical) **but** the problem is **$\mathcal{NP}$-complete in the strong sense**: most likely it will never be possible to solve it in a time that grows $\underline{\text{polynomially}}$ with n.

Hamiltonian circuits

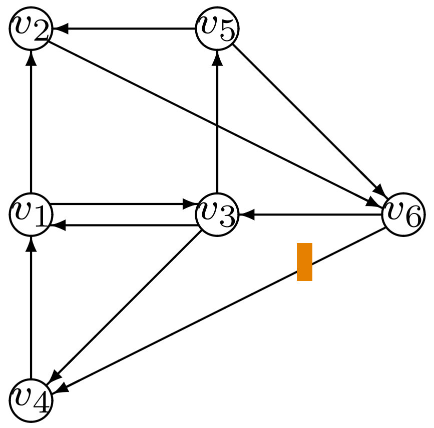
- Given a directed non-weighted graph $G = (V, A)$, does it possess a Hamiltonian circuit (HC)?
- **Implicit Enumeration algorithm**:
- branch-decision tree:

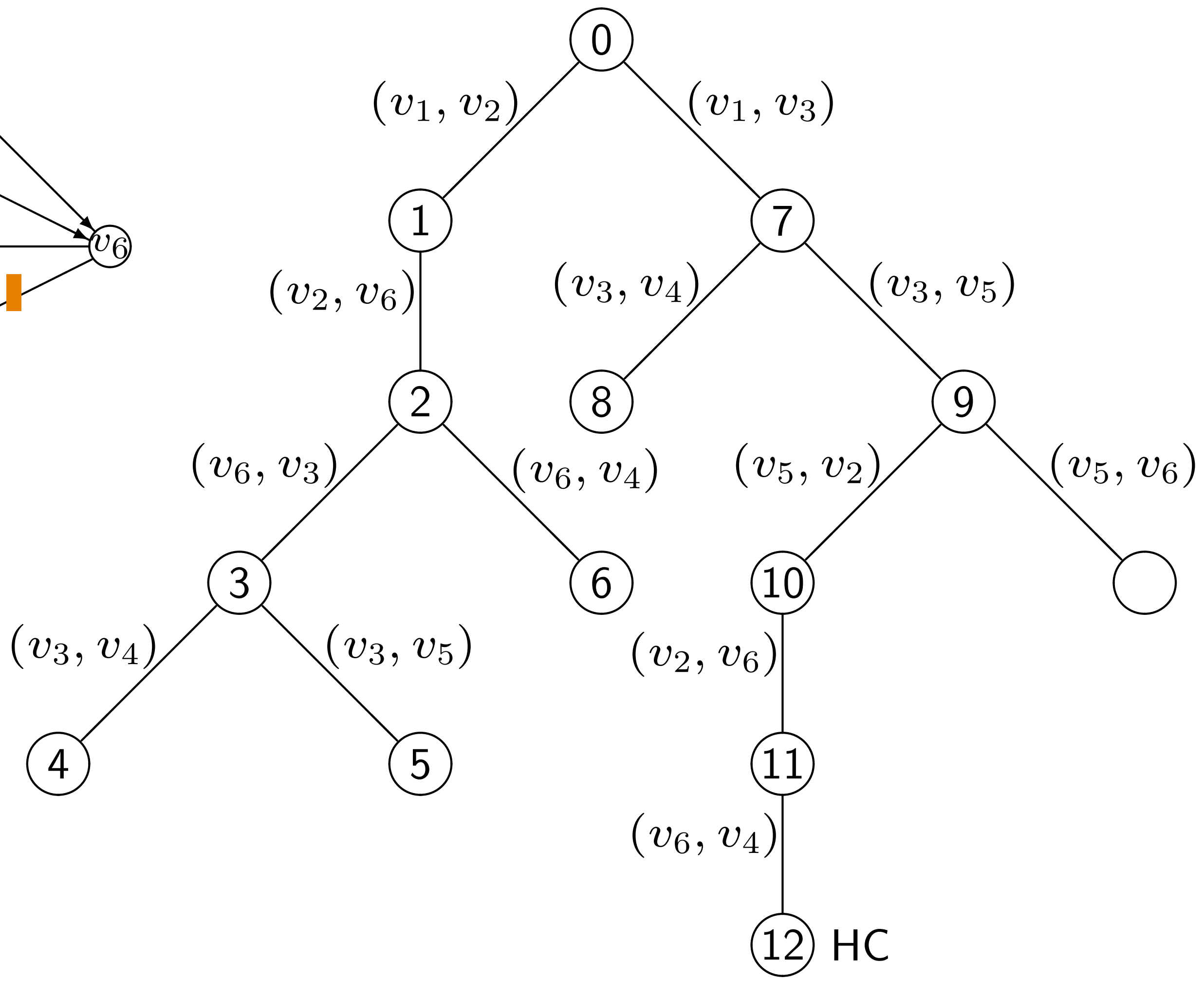


- **Depth-first** search $\Leftrightarrow$ augment a path $S = \{v_1, \dots, v_\ell\}$ until
  - no new vertex can be added to S ($\Rightarrow$ backtracking), or
  - $|S| = n$: **if** $(v_\ell, v_1) \in A$ **then** stop **else** backtrack.
- In the worst case the algorithm takes time $O((n-1)!)$.

Hamiltonian circuits (cont'd)

- **Example:**





Hamiltonian circuits (cont'd)

- To kill decision nodes:

  Observation: when the decision is to add (v_ℓ, v_s) to S, we can remove from A:

  (i) all arcs (v_ℓ, v_i) with $i \neq s$;

  (ii) all arcs (v_j, v_s) with $j \neq \ell$

  (reinserting them when backtracking on (v_ℓ, v_s)).

- When branching on the current path $S = \{v_1, \dots, v_\ell\}$, the following **Rules** can be adopted:

  1. **if** $\exists \ (v_\ell, v_i) \in A, \ v_i \notin S \ : |\Gamma^-(v_i)| = 1$, **then** only the child corresponding to (v_ℓ, v_i) is generated ($\Leftarrow$ otherwise v_i would become unreachable). **Hence**

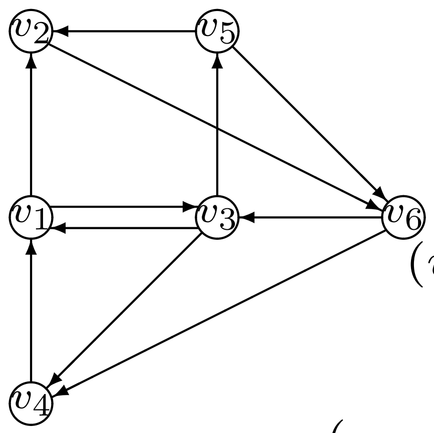
  2. **if** $\exists \ (v_\ell, v_j)$ and (v_ℓ, v_k) like (v_ℓ, v_i) of Rule 1, **then** backtrack.

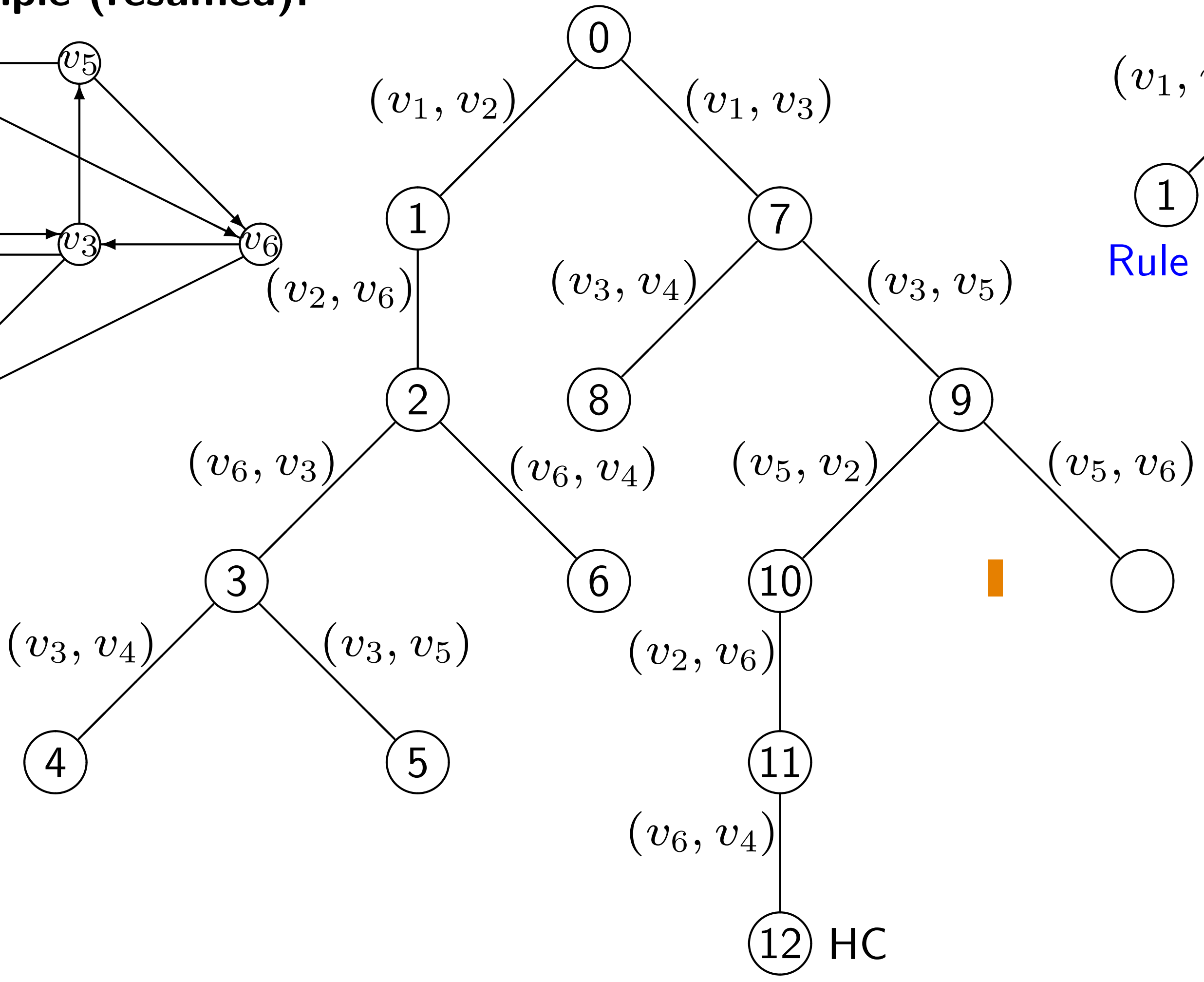
  3. **if** $\exists \ (v_\ell, v_i), (v_k, v_i) \in A, \ v_i, v_k \notin S : |\Gamma^+(v_k)| = 1$, **then** the child corresponding to (v_ℓ, v_i) is not generated (otherwise there would be no continuation from v_k).

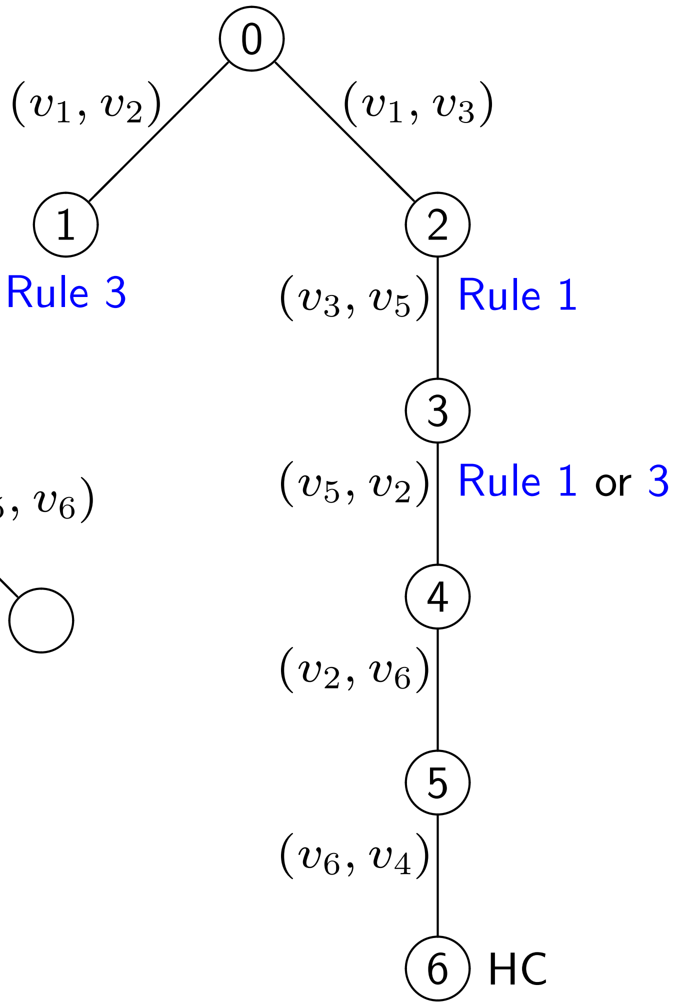
- In the **course web page**: **applet** for executing the enumerative method for the HC.

Hamiltonian circuits (cont'd)

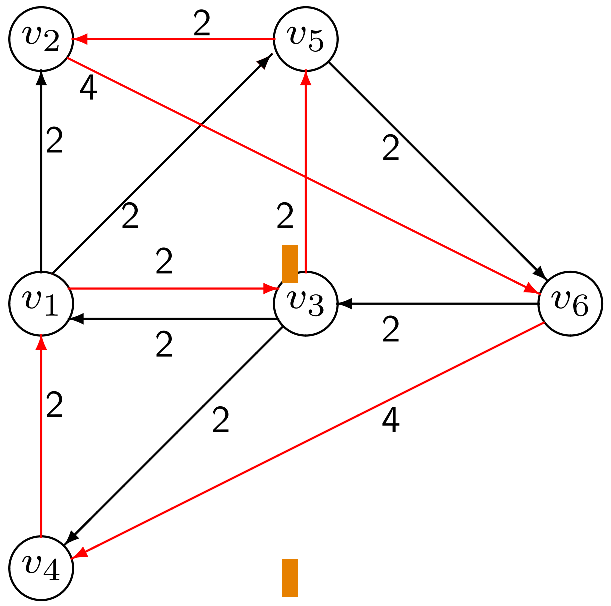
- **Example (resumed):**



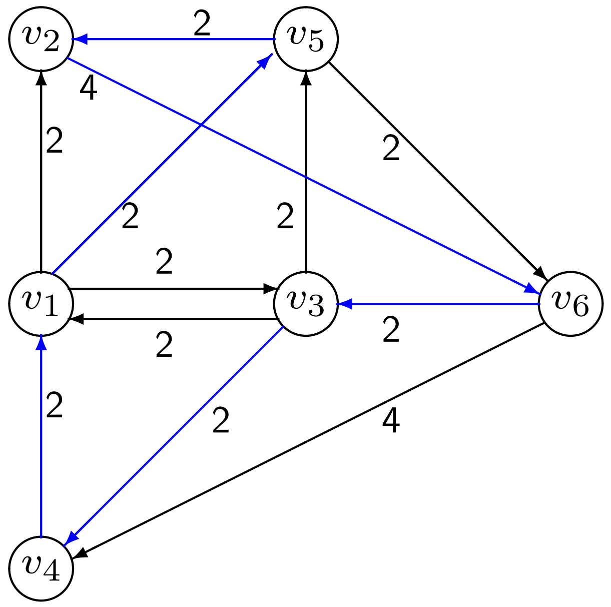




If the graph has **weights** (costs, lengths, times, ...) associated with the arcs/edges, the **Traveling Salesman Problem (TSP)** is to find the **Hamiltonian circuit of minimum total weight**:



cost = 16



cost = 14

Book published in 1931 in German: *The Traveling Salesman Problem, how he should be and what he should do to be successful in his business. By a veteran traveling salesman.*

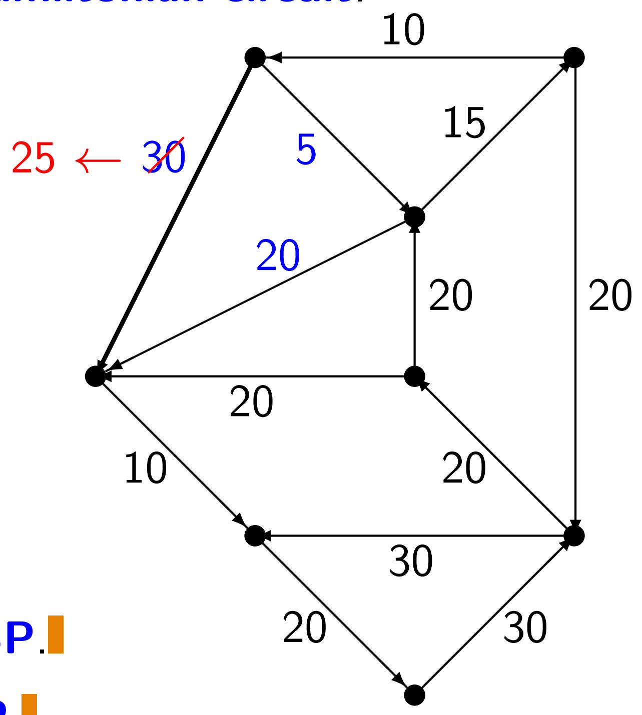
Generalization of the Hamiltonian circuit problem $\Longrightarrow$ **$\mathcal{NP}$-hard in the strong sense**.

Applications in freight transportation using a **single vehicle**. **BUT**

Real world applications use a **fleet of vehicles** based at **one or more depots**, and have **specific constraints**.

Traveling Salesman Problem (TSP)

- Given a graph $G = (V, A)$ (or $G = (V, E)$) having a cost c_{ij} associated with each arc/edge (v_i, v_j), find the **minimum cost Hamiltonian circuit**.



- If the graph is **undirected** $\Rightarrow$ **Symmetric TSP**.
- If the graph is **directed** $\Rightarrow$ **Asymmetric TSP**.
- We assume that:
  1. the **cost matrix is complete**, with $c_{ij} = +\infty$ if arc/edge (v_i, v_j) does not exist.
  2. the **triangle inequality** holds: $c_{ij} \leq c_{ik} + c_{kj}$ $(i, j, k = 1, \ldots, n)$.

     (If it doesn't, set $c_{ij} :=$ *shortest path* from v_i to v_j $\forall i, j$.)

Mathematical model of the Asymmetric TSP

$$x_{ij} = \begin{cases} 1 & \text{if arc } (v_i, v_j) \text{ belongs to the solution} \\ 0 & \text{otherwise} \end{cases} \qquad (i, j = 1, \dots, n)$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \; j = 1, \dots, n \qquad (1)$$

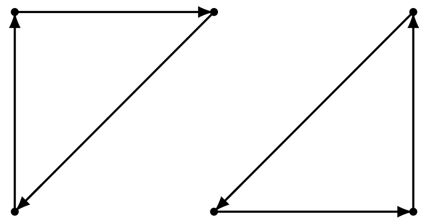
$$\sum_{j=1}^{n} x_{ij} = 1, \; i = 1, \dots, n \qquad (2)$$

Constraints (1): **exactly one arc entering** each vertex;

Constraints (2): **exactly one arc emanating** from each vertex;

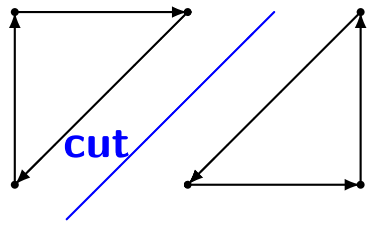
Same model as the Assignment Problem. **Question: is it correct for the Asymmetric TSP?**

Answer: NO! Without additional conditions, **partial circuits** are possible:



Mathematical model of the Asymmetric TSP (cont'd)

Additional constraints: every **cut** $(S, V \setminus S)$ **must be crossed** by at least one arc:



$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\begin{aligned}
\sum_{i=1}^{n} x_{ij} &= 1, \; j = 1, \dots, n \\
\sum_{j=1}^{n} x_{ij} &= 1, \; i = 1, \dots, n \\
\sum_{i \in S, j \notin S} x_{ij} &\geq 1, \; \forall S \subset V, S \neq \emptyset \quad \text{(*)} \qquad (3) \\
x_{ij} &\in \{0, 1\} \; \forall \, (v_i, v_j) \in A
\end{aligned}$$

(*) Alternatively: $\sum_{i \in S, j \in S} x_{ij} \leq |S| - 1, \; \forall S \subset V.$

(*) In both cases, **Exponential number of constraints!**

The assignment problem is used as a relaxation of TSP in branch-and-bound algorithms.

Mathematical model of the Symmetric TSP

$$x_e = \begin{cases} 1 & \text{if } \textbf{edge } e \ (= (i,j) = (j,i)) \text{ belongs to the solution} \\ 0 & \text{otherwise} \end{cases} \qquad (e = 1, \dots, m)$$

$\delta(i) = \{(i,j) \ : \ j \in V\}$ for $i \in V$

$E(S) = \{(i,j) \ : \ i \in S$ **and** $j \in S\}$ for $S \subset V$, $S \neq \emptyset$

$c(e)$ = cost of edge e

$$\min \tfrac{1}{2} \sum_{i=1}^{n} \sum_{e \in \delta(i)} c_e x_e$$

$$\sum_{e \in \delta(i)} x_e = 2, \ i = 1, \dots, n \tag{4}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \ \forall S \subset V, \ S \neq \emptyset \tag{5}$$

$$x_e \in \{0,1\}, \ \forall \ e \in E$$

Constraints (4): **exactly two edges** incident with each vertex;

Constraints (5): no vertex subset can have a **partial circuit** (**Exponential number!**);

Objective function: the cost of each edge (j,l) of the solution is summed **twice** in $\sum_{i=1}^{n} \sum_{e \in \delta(i)} c_e x_e$ (for $i = j$ and for $i = l$) $\Longrightarrow \frac{1}{2}$. Equivalent objective function: $\sum_{e \in E} c_e x_e$.

Vehicle Routing Problems

- Optimal delivering of goods, using a set of **vehicles**, based at one or more **depots**, through a **road network**.
- **Freight transportation**
  - takes 10% - 25% of the final cost for consumer goods;
  - use of optimization techniques $\Longrightarrow$ savings of 5% - 20% of the total transportation costs.
- General **Vehicle Routing Problem (VRP):** ***Find a set of routes, each assigned to a vehicle that starts and ends at its depot, so that a set of specific operational constraints is satisfied and the total transportation cost is minimized.***
- Generalization of the Traveling salesman problem $\Longrightarrow$ $\mathcal{NP}$**-hard in the strong sense**.
- Different models according to the considered constraints. Most models: **undirected graphs**

  Basic (simplest) VRP: **Capacitated Vehicle Routing Problem (CVRP)**:
  - vertex 0=depot; vertices i $(i = 1, \ldots, n)$=customers; edge costs $c_{ij}(i, j = 0, 1, \ldots, n)$;
  - customer i requests goods of total weight d_i $(i = 1, \ldots, n)$;
  - each customer must be visited by a single vehicle;
  - K vehicles having capacity C (or having capacities C_k $(k = 1, \ldots, K)$);
  - the total weight on each vehicle must be $\leq C$ (or on vehicle k must be $\leq C_k \ \forall \ k$).
  - minimize the total cost of the routes.

Vehicle Routing Problems (cont'd)

- **Distance-Constrained CVRP**: like CVRP, with an additional constraint:
  - t_{ij} = traveling time of edge (v_i, v_j);
  - T_k = maximum total traveling time for vehicle k $(k = 1, \dots, K)$.
- **VRP with Time Windows**: like CVRP, with an additional constraint:
  - t_{ij} = traveling time of edge (v_i, v_j);
  - for each customer i,
    * s_i = time needed to download goods;
    * $[a_i, b_i]$ = time window within which the delivery must be made.
- **VRP with Backhauls**: like CVRP, with an additional constraint:
  - the customers are partitioned into two sets:
    * *linehaul customers* to whom goods are to be delivered;
    * *backhaul customers* whose goods need to be transported back to the distribution center;
  - each vehicle must visit all its linehaul customers before all its backhaul customers;
  - for each vehicle k, $\max\{$(total linehaul weight),(total backhaul weight)$\} \le C_k$.

Vehicle Routing Problems (cont'd)

- **VRP with Pickup and Delivery**: like CVRP, with an additional constraint:
  - for each customer i:
    - d_i = quantity of goods to be delivered to the customer;
    - p_i = quantity of goods to be picked up at the customer;
  - when the vehicle arrives, it first downloads then uploads;
  - for each vehicle k the total carried weight at any time must be $\leq C_k$.
- **VRP with Loading Constraints**: like CVRP, with an additional constraint:
  - for each customer i:
    - list of the packages to be delivered, with the corresponding dimensions (height, width, depth);
  - for each vehicle k:
    - dimensions of the loading area (height, width, depth);
  - the solution must provide, for each vehicle, the packing in the loading area.
- **Combinations of the various constraints** (e.g., backhauls + time windows + ...)
- **Variants** (multiple depots, customers served by more vehicles, special compartments, ...)