Pina Kolling
piko0011

Summary: Lecture 8

Summary for the chapters *9.3* and *9.4*. [2, 1]

Undirected graph

An undirected graph G is a pair of sets (V, E) where V is the finite set of nodes and E is a set of unordered pairs in V that are symmetric:

$$\forall\ i, j \in E, i \neq j :\ (i, j) \in E \Rightarrow (j, i) \in E$$

IndependentSet

IndependentSet

INDEPENDENTSET
Input: An undirected Graph $G = (V, E)$ and a number k.
Question: Is there a set $I \subseteq V$ of $k = |I|$ nodes with no edges in between? (INDEPENDENTSET)

INDEPENDENTSET is NP-complete.

Proof idea:

- triangle construction: any independent set can contain at most one node of the triangle

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

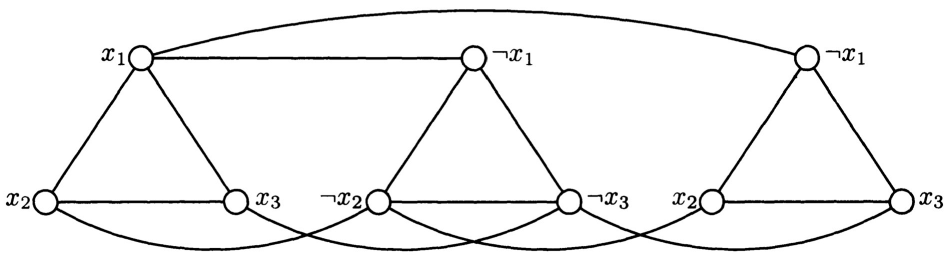

Figure 1: Graph with triangles [2]

- consider only graphs whose nodes can be partitioned in m disjoint triangles
  $\rightarrow$ independent set can contain at moast m nodes (one from each triangle)
- reduction from 3SAT to INDEPENDENTSET
- construct graph of formula ϕ:
  - each literal as a node
  - clauses as triangles
  - edges between nodes in different triangles if they correspond to the same literal (negated)
  - $K = m$ (m clauses)
- given: instance ϕ of 3SAT with m clauses $C_1, ..., C_m$
- each clause $C_i = (\alpha_{i1} \vee \alpha_{i2} \vee \alpha_{i3})$ (with α as boolean variables or negation of those)
- reduction R constructs a graph: $R(\phi) = (G, K)$ where $K = m$ and $G = (V, E)$

- nodes $V = \{v_{ij} : \ i = 1, ..., m; j = 1, 2, 3\}$
  nodes for each of the m clauses (i) for each of the 3 literals (j)
- edges $E = \{[v_{ij}, v_{ik}] : \ i = 1, ..., m; j \neq k\} \cup \{[v_{ij}, v_{lk}] : \ i \neq l, \alpha ij = \neg\alpha_{lk}\}$
  edges between the nodes in one clause (triangle edges)
  edges between nodes with the same corresponding literal, but negated

TODO
Questions:

HamiltonPath is NP-complete

HamiltonPath is NP-complete.

Proof idea:

- Another reduction from 3Sat

TODO
Questions:

TSP(D)

TSP(D)
TSP(D) is a decision version of TSP. Input: A $n \times n$ distance matrix and a bound $B \in \mathbb{N}$ Question: Is there a round tour of length $\leq B$ that visits all *cities*?

TSP(D) is NP-complete.

Proof idea:

- budget of nodes is $B = |V| + 1$

TODO
Questions:

Knapsack

Knapsack

Knapsack is NP-complete.

- filled in in one dimensional array onthe board
-

TODO
Questions:

References

[1] Martin Berglund. *Lecture notes in Computational Complexity*.

[2] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, 1994.