Polynomial time algorithms

Examples Sorting, Kruskal, Dijkstra

Common feature: number of steps bounded by a polynomial of the input size.

Finishing in polynomial time is the theoretical choice for an algorithm being "fast".

- Ignores running time for small input size, concentrates on performance on large input; not necessarily fast in practice
- But (unpractical) theoretical advance often inspires improvements in practice
- Can take advantage of improving technology more
- Constant factor: hides dependence on concrete system/implementation
- Less sensitive to definition of operation in runtime calculations

Comparing $O(n^{100})$ and $O(2^n)$ algorithms

The class P

We restrict to dealing with *decision problems* (problems with a YES or NO answer to every input)

The class P consists of decision problems that can be solved by a polynomial time algorithm

Optimization problems can also be formulated as decision problems

Arthur and Merlin – a glimpse of complexity_

A: Show me a pairing, so my 150 knights can marry these 150 ladies!
M: Not possible!
A: Why?
M: Here are these 93 ladies and 58 knights, none of them are willing to marry each other.
A: Alright, alright ...

A: Seat my 150 knights around the round table, so that neighbors don’t fight!
M: Not possible!
A: Why?
M: It will take me forever to explain you.
A: I don’t believe you! Into the dungeon!

The class NP

A YES/NO-problem is in the class *NP*, if for any YES-input it could be verified fast (in time polynomial in the input size), that the answer to the input is indeed **YES**.

In other words: there is a ”certificate”, which a computer (i.e., Arthur, i.e., a polynomial time algorithm) can verify fast (the certificate can be provided by an all-powerful oracle (i.e., Merlin))

NP: “nondeterministic polynomial time”

Examples:

- “Does this bipartite graph have a perfect matching?” (provide perfect matching)
- “Does this bipartite graph have **no** perfect matching?” (provide a subset S of one of the sides that has less neighbors than elements; such a certificate exists by Hall’s Theorem)
- “Does this graph have a Hamilton cycle?” (provide Hamilton cycle)

Merlin’s Pech: “Does this graph have **no** Hamilton cycle?” is not (known to be) in NP

P vs NP

Recall: a YES/NO-problem is in the class *P*, if there is polynomial time algorithm that **decides** whether the answer is YES or NO.

Of course: $P \subseteq NP$
(the running of the algorithm is itself a certificate (i.e., no need for Merlin))

Conjecture. $P \neq NP$ (1,000,000 US dollars)

Potential examples in $NP \setminus P$: Hamiltonicity, 3-colorability, independence number, . . .

NP-hard problem: every problem in NP can be reduced to it in polynomial time (consequently, giving a polynomial time algorithm for it would result in a polynomial time algorithm for **all** problems in NP, and hence P=NP)

NP-complete problem: NP-hard and contained in NP

Many problems are NP-complete: Hamiltonicity, 3-colorability, 4-colorability, planar 3-colorability
(but 2-colorability and planar 4-colorability are in P)

Characterizations

A YES/NO-problem is in the class *co-NP:* The answer **NO** can be verified efficiently

Property has a “good” characterization $\rightarrow$ corresponding decision problem is both in NP and co-NP

Examples:
- Does this bipartite graph have a perfect matching?
- ”Is this graph 2-colorable?” (NP-certificate: 2-coloring; co-NP-certificate: odd cycle)
- ”Is this graph Eulerian?” (NP-certificate: ordered list of the edges for an Eulerian circuit; co-NP-certificate: vertex with odd degree (**exists** because of Euler’s Theorem))

Of course: $P \subseteq NP \cap co\text{-}NP$

Often: Problems in $NP \cap co\text{-}NP$ are also in P

However: People think $P \neq NP \cap co\text{-}NP$

We don’t know: status of problem ”Is there a factor of n less than k?” (until 2002 even the status of the problem “Is n a prime?” was also not known)

Perfect matchings in general graphs

NP-co-NP-characterization for bipartite graphs:

Corollary (of Hall's Marriage Theorem) There is a perfect matching in a bipartite graph $G = (X \cup Y, E)$ iff $|X| = |Y|$ and $|N(S)| \geq |S|$ for every $S \subseteq X$.

Towards an NP-co-NP-characterization for general graphs. Obstructions for p.m.: no graph on odd vertices has p.m. The disjoint union of two odd graphs has even number of vertices, but no p.m. The disjoint union of three odd graphs with each connected to a single vertex is connected, has even number of vertices, but no p.m. Etc ...

An odd component is a connected component with an odd number of vertices. Denote by $o(H)$ the number of odd components of a graph H.

Theorem. (Tutte, 1947) A graph G has a perfect matching iff $o(G - S) \leq |S|$ for every subset $S \subseteq V(G)$.

Corollary. Decision problem "Does this graph have p.m.?" is in NP∩co-NP.

Matchings in general graphs

Proof.

$\Rightarrow$ Easy (think over).

$\Leftarrow$ (Lovász, 1975) Consider a counterexample G (G satisfies Tutte's Condition, but has no perfect matching) with the maximum number of edges.

Claim. $G + xy$ has a perfect matching for any $xy \notin E(G)$.

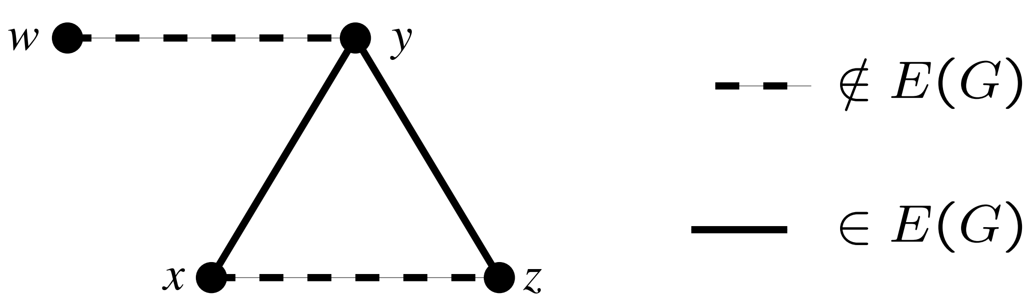
Define $U := \{v \in V(G) : d_G(v) = n(G) - 1\}$

Case 1. $G - U$ consists of disjoint cliques.

Straightforward to construct a perfect matching of G (using Tutte's condition for $S = \emptyset$). Contradiction.

Case 2. $G - U$ is not the disjoint union of cliques.

Derive the existence of four vertices x, y, z, w such that $yx, yz \in E$ and $xz, yw \notin E$.



Idea: Obtain contradiction by constructing a perfect matching M of G using perfect matchings M_1 and M_2 of $G + xz$ and $G + yw$, respectively, and the edges yx and yz.

How? $M_1 \cup M_2$ is a subgraph of $G + \{xz, yw\}$ that spans all vertices and all its components are even cycles or single edges. Choose edges of M these component-wise.

If a component C

- does not contain xz: use edges of M_1 to saturate $V(C)$.
- does not contain yw: use edges of M_2 to saturate $V(C)$.
- contains both xz and yw: Then C is an even cycle and the path $C - y$ is of even length. x and z are neighbors on $C - y$ and not endpoints, so the removal of one of them cuts $C - y$ into two paths of odd length. On each of these odd paths take the unique maximum matching and saturate the remaining two vertices of C (that were removed) by the edge of G connecting them. (yx or yz)

Perfect matchings in regular graphs

Corollary of Hall's Theorem: Every k-regular bipartite graph, $k \geq 1$, has a perfect matching

2-regular non-bipartite graph might have no perfect matching. (odd cyces)

BUT! Corollary of Tutte's Theorem:

Theorem. (Petersen, 1891) Every 3-regular graph with no cut-edge has a perfect matching.

Proof. Check Tutte's condition. Let $S \subseteq V(G)$.
Double-count the number of edges between an S and the odd components of $G - S$.
Observe that between any odd component and S there are at least three edges.

Failed tries for characterizing Hamiltonicity

Recall: a sufficient condition from DMI.

Dirac's Theorem. If $G = (V, E)$ is a simple graph on $n \geq 3$ vertices and $\delta(G) \geq \frac{n}{2}$, then G is Hamiltonian.

The condition is best possible, but not necessary (C_n)

A slightly weaker sufficient condition:

Ore's Condition. Let G be n vertex graph such that for every $uv \notin E(G)$, we have $d(u) + d(v) \geq n$, then G is Hamiltonian.

Still not necessary...

A necessary condition? What is true for a single Hamilton cycle?

Proposition. If G is Hamiltonian, then for every $S \subseteq V$, $c(G - S) \leq |S|$ (where $c(H)$ is the number of components of graph H).

This is not sufficient. (Petersen graph)

Trying to strengthen the sufficient condition. A graph G is t-tough if $|S| \geq tc(G - S)$ for every cut-set $S \subseteq V(G)$. The toughness of G is the maximum t such that G is t-tough.

The toughness of the Petersen graph is $4/3$.

Toughness Conjecture (Chvátal, 1973) There is a value t such that every graph of toughness at least t is Hamiltonian.

Bauer-Broersma-Veldman (2000) constructed a family of non-Hamiltonian graphs with toughness approaching $9/4$. So the conjecture is **not true** for $t < \frac{9}{4}$.

And even if we were able to prove the Toughness Conjecture for some t, it would not give a characterization of Hamiltonicity, only sandwich it between two properties that are a "constant factors away from each other".

Maximum matching problem

Optimization version of matching problem: “What is the **size** of the maximum matching?”

Decision problem: “Is the maximum matching in this graph is at least k?”

Perfect matching problem is special case: $k = \frac{n}{2}$.

Is the maximum matching problem also in NP∩co-NP for every (function) $k = k(n)$?

Our NP-co-NP-characterization theorems turn out to have the appropriate generalizations. $\rightarrow$ Min-Max Theorems.

Certificate for bipartite graphs — Take 1

How to convince Arthur that in a bipartite graph $G = (X \cup Y, E)$ there is no matching of size larger than k?

Find a subset $S \subseteq X$, such that $k = |X| - |S| + |N(S)|$

1. Correctness of the certificate:

For any matching M and subset $S \subseteq X$, at least $|S| - |N(S)|$ vertices of S are not saturated by M (since vertices of S can only be matched into distinct vertices of $N(S)$).

2. Existence of optimal certificate:

By a min-max generalization of Hall's Theorem:

Hall's Theorem (Min-max version) For every bipartite graph $G = (X \cup Y, E)$,

$$\alpha'(G) = \min\{|X| - |S| + |N(S)| : S \subseteq X\},$$

where $\alpha'(G) =$ size of largest matching.
Proof. HW

This implies that there exists a subset $S \subseteq X$, such that $|X| - |S| + |N(S)| = \alpha'(G)$.

Certificate for bipartite graphs — Take 2

Recall: $C \subseteq V(G)$ is a vertex cover if for every edge $e \in E(G)$, $e \cap C \neq \emptyset$.

$\beta(G) = \min\{|C| : C \text{ is a vertex cover}\}$

To convince Arthur that in an arbitrary(!) graph there is no matching of size larger than k, it is enough to exhibit a vertex cover of size k.

1. Correctness of the certificate:

For any vertex cover $Q \subseteq V(G)$ and matching $M \subseteq E(G)$, every $e \in M$ must contain *at least one* vertex of Q and these are all distinct. $\Rightarrow$ $|Q| \geq |M|$ (and hence $\beta(G) \geq \alpha'(G)$ holds for every graph G).

2. Existence of optimal certificate for bipartite graphs:

Theorem. (König (1931), Egerváry (1931))
If G is bipartite then $\beta(G) = \alpha'(G)$.

Proof. For any minimum vertex cover Q, apply Hall's Condition to match $Q \cap X$ into $Y \setminus Q$ and $Q \cap Y$ into $X \setminus Q$. $\square$

König's Theorem $\Rightarrow$ For bipartite graphs there always exists a vertex cover of size $\alpha'(G)$, convincing Arthur that a particular matching of maximum size is really maximum.

Remark. Such a cover NOT necessarily exists for non-bipartite graphs (for example, odd cycles).

Corollary The problem "Is there a matching of size k in this bipartite graph?" is in NP∩co-NP, for any (function) $k = k(n)$.

Certificate for arbitrary graphs

To convince Arthur that in a graph there is no matching of size larger than k, it is enough to exhibit a subset $S \subseteq V$ such that $2k = n - (o(G-S) - |S|)$.

Correctness of certificate: For any subset $S \subseteq V$, any matching M, and any odd component C of $G-S$, *at least* one vertex of C is not saturated by an M-edge *within* C. Since these vertices can then only be connected by an M-edge to vertices of S, all distinct(!), at least $o(G-S) - |S|$ vertices in odd components are not saturated by M.

Existence of certificate:
Tutte's Theorem (Min-max version) (Berge)
In every graph G, the maximum number of vertices saturated by a matching is

$$2\alpha'(G) = \min\{n - o(G-S) + |S| : S \subseteq V(G)\}.$$

Proof. HW

Corollary The problem "Is there a matching of size k in this graph?" is in NP∩co-NP, for any (function) $k = k(n)$.

Approximation algorithm for TSP

One way to get closer to solving an NP-complete problem of the optimization kind: give fast algorithm that finds a solution *as close to optimal as you can*.

How light is the lightest Hamilton cycle in a graph with given edge weights?

Recall: **Traveling Salesman Problem (TSP)**

Given a weight function $w : E(K_n) \to \mathbb{R}_{\geq}$ on the edges, find a Hamilton cycle H of smallest weight $w(H) = \sum_{e \in E(H)} w(e)$.

Special case: Is there a Hamilton cycle in a graph G? (reduction via $1/2$-weights)

Hence (the decision problem version of) TSP is NP-complete as well.

A practical approach: Let w_{OPT} be the weight of a Hamilton cycle of minimum weight. For a $c \geq 1$, a c-*approximation algorithm* is an algorithm which outputs a Hamilton cycle H with $w(H) \leq c \cdot w_{OPT}$

Algorithm `TSP-Approx`

Step 1. Find MST T

Step 2. Create walk W “around” T, traversing each edge twice

Step 3. Set $H = W$ and go around H and iteratively change it by “shortcuting” at any vertex which is used the second time. Output H when $e(H) = n$

Remark. Running time: fast (Kruskal + $O(n)$)

Theorem If w satisfies the triangle inequality, then `TSP-Approx` is a 2-approximation algorithm.

Proof. Let T_{min} a MST of G. Then $w(W) = 2w(T_{min})$

By triangle inequality, shortcut decreases the sum of the weights of H, so $w(H) \leq 2w(T_{min})$

Hamilton path within an optimal traveling salesman tour is a spanning tree, so $w(T_{min}) \leq w_{OPT}$ □