Advanced algorithms

Exercise sheet #2 (Solutions) – NP-completeness, branch-and-bound, exponential-time algorithms

September 28, 2022

Exercice 1 (NP-completeness of HITTING SET)**.** The problem HITTING SET is the following:

Input: A finite collection $\mathcal{C}$ of finite sets of integers; an integer $k \geqslant 0$.
Question: Is there a set $S \subset \mathbb{N}$ such that $\#S \leqslant k$ and $A \cap S \neq \varnothing$ for any $A \in \mathcal{C}$?

Using the NP-complete problem VERTEX COVER, show that HITTING SET is NP-complete.

Solution — It is clear that HITTING SET *is in NP because the set S is a certificate that we can check in polynomial time.*

Let $G = (V, E)$ a graph and $k \geqslant 0$ an integer, forming an instance of VERTEX COVER*. We may assume that $V \subset \mathbb{N}$, i.e. the vertices are integers. Let $\mathcal{C} = E$. It is easy to check that G has a vertex cover with $\leqslant k$ elements if and only if $\mathcal{C}$ has a hitting set with $\leqslant k$ elements. Since* VERTEX COVER *is NP-hard, then* HITTING SET *is also NP-hard. Since it is in NP, it is also NP-complete.*

Exercice 2 (NP-completeness of FEEDBACK ARC SET)**.** Let $G = (V, E)$ be a directed graph. A *feedback arc set* is a subset of E such that the graph $(V, E \setminus C)$ is acyclic.

The problem FEEDBACK ARC SET is the following:

Input: A directed graph G; an integer $k \geqslant 0$.
Question: Does G admit a feedback arc set of cardinality $\leqslant k$?

(a) Show that FEEDBACK ARC SET is in NP.

Solution — We can check that a given set of edges is a feedback arc set in polynomial time using a depth first search to detect cycles.

Given an undirected graph $G = (V, E)$, we define a directed graph $G' = (V', E')$ as follows: $V' = V \times \{0, 1\}$ and $E' = \left\{(x^0, y^1) \text{ s.t. } \{x, y\} \in E\right\} \cup \left\{(x^1, x^0) \text{ s.t. } x \in V\right\}$, where, for short, we denote x^i the pair $(x, i) \in V'$.

(b) Show that G' has a minimum-cardinality feedback arc set with edges only of type (x^1, x^0).

Solution — Let C be a feedback arc set of G'. Replace each edge (x^0, y^1) in C by (y^1, y^0) to form a set C'. Then $|C'| \leq |C|$. Assume that there exists a cycle γ in $G'' = (V', E' \backslash C')$. Since $(V', E' \backslash C)$ is acyclic, γ must pass through an edge $c \in C$, which is of the form (x^0, y^1). The following edge in γ is then (y^1, y^0), which belongs to C'. Contradiction. Therefore, G'' is acyclic, and C' is a feedback arc set of G'.

(c) Show that Feedback arc set is NP-complete. *Hint: use* Vertex cover.

Solution —

Let $(G = (V, E), k)$ be an instance of vertex-cover*,* i.e. *G is an undirected graph and k is some non negative integer.*

Let $G' = (V', E')$ be the directed graph obtained via the previous construction.

- *Assume that G admits a vertex cover S of size $\leq k$, and let $F_S := \{(x^1, x^0) \mid x \in S\}$. If γ is a cycle of $(V', E' \backslash F_S)$ then it passes through an edge of the form (x^0, y^1), and therefore through (x^1, x^0) and (y^1, y^0). It corresponds to the edge $\{x, y\} \in E$. Since S is a vertex cover, we can assume without loss of generality (wlog) that $x \in S$. But then, $(x^1, x^0) \in F_S$ which contradicts the fact that γ passes through it. So F_S is a feedback arc set of size k.*
- *Assume that G' admits a feedback arc set C of size $\leq k$. Using the previous question, we can assume wlog that C contains only edges of type (x^1, x^0). Let $S_C := \{x \mid (x^1, x^0) \in C\}$. Let $\{a, b\} \in E$. Notice that $(a^1, a^0), (a^0, b^1), (b^1, b^0), (b^0, a^1)$ is a cycle of G'. Therefore, either $(a^1, a^0) \in C$ or $(b^1, b^0) \in C$, which is exactly $a \in S_C$ or $b \in S_C$,* i.e. *S_C is a vertex cover.*

We proved that G has a vertex cover of size $\leq k$ if and only if G' has a feedback arc set of size $\leq k$. Since constructing G' can be done in polynomial time, Feedback arc set *is NP-complete.*

Exercice 3 (Dynamic programming for TSP)**.** Let $G = (V, E)$ be a complete graph with a weight function $w : E \to \mathbb{R} \cup \{+\infty\}$. The *travelling salesman problem (TSP)* is the problem of computing a minimum-weight cycle in G that goes through all vertices exactly once. The naive solution to solve this problem is the enumeration of all $n!$ cycles (where $n = \#V$).

The goal of this exercise is to do better and to provide an exponential time algorithm.

(a) Let $S \subset V$ containing at least two vertices, and let $s \in S$. For any $t \in S$, we denote by $W(S, t)$ the total weight of a shortest path from s to t, visiting exactly once the elements of S and no other vertex.

Give a recursive expression of $W(S, t)$.

Solution — Let γ be a path reaching the optimum $W(S, t)$, and let a be the penultimate vertex in γ. Therefore, the subpath from s to a reaches the optimum $W(S \backslash \{t\}, a)$. Then,

$$W(S, t) = \min_{a \in S \backslash \{t\}} \Big(W(S \backslash \{t\}, a) + w(a, t) \Big).$$

(b) Deduce an algorithm to solve TSP in time $O(n^2 2^n)$ using the principle of *dynamic programming*.

Solution — Fix a vertex $s \in V$ as starting and ending point. By definition, and using the notations of the previous question, the minimum-weight solution of TSP on G is given by

$$\min_{t \in V} \Big(W(V, t) + d(t, s) \Big).$$

For any t, we can compute recursively $W(V,t)$ by dynamic programming using the previous question: It suffices to compute all the $W(S,t)$ where $S \subset V$, starting from $\#S = 2$ and increasing S. For each t, there are 2^n subsets $S \subset V$ and therefore there are $n \times 2^n$ such pairs (S,t).

Notice that for any vertex $a \in V$, $W(\{a,s\},a) = w(a,s)$, and using the previous formula we can compute $W(S,t)$ by induction on $\#S$, wich costs $O(n)$ operations.

Therefore, the solution of TSP can be computed in time $O(n^2 2^n)$.

Exercice 4 (WalkSAT)**.** We study a randomized local search procedure to solve 3-SAT problem. It takes as input a 3-SAT instance P and an assigment a of the variables that does not satisfy P:

procedure Update(P, a)
  Choose a clause c of P that is not satisfied by a
  Choose randomly one the variables appearing in c
  Flip the assignment of v in a
end procedure

For two affectations a and b of the variables of P, let $d(a,b)$ denote the Hamming distance of a and b: this is the number of variables that are not assigned to the same value in a and b.

We consider now a 3-SAT instance P that has a solution s.

(a) Let a be an assignment of the variables of P that does not satisfy P, and let $a' =$ Update(P,a). Show that $d(a',s) = d(a,s) \pm 1$ and that

$$\mathbb{P}\left[d(a',s) = d(a,s) - 1\right] \geqslant \frac{1}{3}.$$

Solution — Let c be a clause of P that is not satisfied by a, that is, all three literals in c are given the value 0 by a. The clause c is satisfied by s, so at least one of the literals is given the value 1 by s. Let v be the variable in this literal.

The value of v in a is the opposite of its value in s, otherwise c would be satisfied by a. With probability $\frac{1}{3}$, the variable v is chosen to be flipped. In this case $d(a',s) = d(a,s) - 1$.

We now consider the following algorithm that applie the procedure Update until finding a solution, or stops with the symbol $\varnothing$ if it has not found anything after N iterations.

Intput: a 3-SAT instance P, an assignment a and an integer N
Output: a solution of P or $\varnothing$

procedure WalkSAT(P, a, N)
  for k from 1 to N **do**
    if a satisfies P **then**
      return a
    end if
    $a \leftarrow$ Update(P,a).
  end for
  return $\varnothing$
end procedure

Let $h(a,N)$ be the probability that WalkSAT(P,a,N) returns a solution of P.

(b) Let a be an assignment of the variables and let $\delta = d(a, s)$. Show that

$$h(a, 3\delta) \geqslant \binom{3\delta}{\delta} \left(\frac{1}{3}\right)^{2\delta} \left(\frac{2}{3}\right)^{\delta}.$$

For this, one can compare $h(a, 3\delta)$ with the probability to obtain δ tails and 2δ faces out of 3δ coin flips with a biased coin.

Solution —

Let a_i denote the assignment of the variables after step i, with $a_0 = a$, and let $D_i := d(a_i, s)$. Using the previous question, D_i realizes a biased random walk on $\mathbb{N}$, with an absorbing state at 0, starting at δ and going to the left with probability $q \geq \frac{1}{3}$ and to the right with probability $1 - q$. Then $h(a, 3\delta)$ is exactly the probability that this random walk reaches 0 in 3δ steps.

We will use a classical proof technique in probability theory called coupling: To compare two random distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ it suffices to introduce a random vector (X, Y) whose marginal distributions are $\mathcal{D}_1$ and $\mathcal{D}_2$ but such that X and Y are related such that it is easy to compare them.

Let (δ_i) be a sequence of i.i.d random variables in $\{-1, 1\}$, such that $\mathbb{P}(\delta_i = -1) = q$ and $\mathbb{P}(\delta_i = 1) = 1 - q$, and let's consider the following random walk on $\mathbb{Z}$:

$$X_0 = \delta \text{ and } X_{i+1} = X_i + \delta_i.$$

Then, the sequence (D'_i) such that $D'_i = X_i$ while $X_i \geq 0$ and 0 once $X_i < 0$ has the same distribution as (D_i) and clearly

$$h(a, 3\delta) = \mathbb{P}(D_{3\delta} = 0) = \mathbb{P}(D'_{3\delta} = 0) \geq \mathbb{P}(X_{3\delta} = 0)$$

and it suffices to find a lower bound on the probability that X reaches 0. To handle that, we will compare this walk to a random walk whose behaviour is well understood.

Now, let (ε_i) be a sequence of i.i.d random variables in $\{-1, 1\}$, such that $\mathbb{P}(\varepsilon_i = -1) = \dfrac{1}{3}$ and $\mathbb{P}(\varepsilon_i = 1) = \frac{2}{3}$, and let's consider the following "perfectly biased" random walk on $\mathbb{Z}$:

$$Y_0 = \delta \text{ and } Y_{i+1} = Y_i + \varepsilon_i.$$

Define a new random walk Z such that $Z_0 = Y_0 = \delta$

- *If $Y_i = -1$ then $Z_i = -1$,*
- *If $Y_i = 1$ then $Z_i = -1$ with probability $\frac{q-1/3}{2/3} = \frac{3q-1}{2}$.*

Then,

$$\begin{aligned}
\mathbb{P}(Z_i = -1) &= \mathbb{P}(Z_i = -1 \mid Y_i = -1)\mathbb{P}(Y_i = -1) + \mathbb{P}(Z_i = -1 \mid Y_i = 1)\mathbb{P}(Y_i = 1) \\
&= 1 \times \frac{1}{3} + \frac{3q-1}{2} \times \frac{2}{3} \\
&= q
\end{aligned}$$

Therefore, the sequence of Z_i has exactly the same distribution of the walk (X_i), but now it is easy to compare it to the walk (Y_i): We clearly have

$$(Y_{3\delta} = 0) \subset (Z_{3\delta} = 0).$$

All in all,

$$h(a, 3\delta) \geq \mathbb{P}(X_{3\delta} = 0) = \mathbb{P}(Z_{3\delta} = 0) \geq \mathbb{P}(Y_{3\delta} = 0).$$

*Conditioning on the number of steps on the right (*i.e. *such that $\varepsilon_i = 1$), we find that*

$$\mathbb{P}(Y_{3\delta} = 0) = \sum_{k=0}^{\delta} \binom{3\delta}{k} \left(\frac{1}{3}\right)^{3\delta-k} \left(\frac{2}{3}\right)^{k} \geq \binom{3\delta}{\delta} \left(\frac{1}{3}\right)^{2\delta} \left(\frac{2}{3}\right)^{\delta}.$$

(c) We admit the lower bound $\binom{3\delta}{\delta} \geqslant (3\delta+1)^{-1} \left(\frac{27}{4}\right)^{\delta}$. If a is random and uniformly distributed among all possible assignments of the n variables of P, show that

$$\mathbb{P}\left[\text{WALKSAT}(P, a, 3n) \text{ returns a solution of } P\right] \geqslant (3n+1)^{-1} \left(\frac{3}{4}\right)^{n}.$$

Solution — For any assignment a, $h(a, 3n) \geqslant h(a, 3\delta)$. Moreover, there is 2^n possible assignments of the n variables, and, for a given δ, there is $\binom{n}{\delta}$ assignments a with $d(a, s) = \delta$. Thus,

$$\begin{aligned}
&\mathbb{P}[\text{WALKSAT}(P, a, 3n) \textit{ returns a solution of } P] \\
&\qquad = 2^{-n} \sum_{a} h(a, 3n) \\
&\qquad = 2^{-n} \sum_{\delta=0}^{n} \sum_{\substack{a \\ d(a,s)=\delta}} h(a, 3n) \\
&\qquad \geqslant 2^{-n} \sum_{\delta=0}^{n} \sum_{\substack{a \\ d(a,s)=\delta}} h(a, 3\delta) \\
&\qquad \geqslant 2^{-n} \sum_{\delta=0}^{n} \sum_{\substack{a \\ d(a,s)=\delta}} \binom{3\delta}{\delta} \left(\frac{1}{3}\right)^{2\delta} \left(\frac{2}{3}\right)^{\delta} \\
&\qquad = 2^{-n} \sum_{\delta=0}^{n} \binom{n}{\delta} \binom{3\delta}{\delta} \left(\frac{1}{3}\right)^{2\delta} \left(\frac{2}{3}\right)^{\delta} \\
&\qquad \geqslant 2^{-n} \sum_{\delta} \binom{n}{\delta} \frac{2^{-\delta}}{3\delta+1} \\
&\qquad \geqslant \frac{1}{3n+1} \left(\frac{3}{4}\right)^{n}.
\end{aligned}$$

(d) Deduce a probabilistic algorithm that returns with probability $\geqslant \frac{1}{2}$ a solution of satisfiable 3-SAT instance with n variables with $\text{poly}(n) \left(\frac{4}{3}\right)^{n}$ operations.

Solution — Call $\textsc{WalkSAT}(P, a, 3n)$ *with independent and uniformly distributed assignments a until obtaining a solution. After* $K = (3n+1)\left(\frac{4}{3}\right)^n$ *calls, the probability to get a solution is at least*

$$1 - (1 - 1/K)^K \geqslant 1 - e^{-1} \geqslant \frac{1}{2}.$$