Web Algorithms

Eng. Fabio Persia, PhD

Optimization problems

- Def: An optimization problem π is a quadruple $(I_\pi, S_\pi, m_\pi, goal_\pi)$ with:
  - I_π set of input instances of π;
  - $S_\pi(x)$ set of feasible solutions of instance $x \in I_\pi$
  - $m_\pi(x,y)$ (integral) measure of feasible solution $y \in S_\pi(x)$ for input $x \in I_\pi$
  - $goal_\pi \in \{min, max\}$ specifies whether we have a minimization or maximization problem

Remarks

- We assume that $m_\pi(x,y)$ is always an integer number
  - our computational models can only deal with rational approximation of reals
  - Scaling such reals we can get equivalent integer numbers
  - Integer values already reveal the intrinsic difficulties of the problems

- When clear from the context in the sequel
  - π will be omitted
  - $m(x,y)$ will be denoted simply as m.

Example: definition of the Clique optimization problem

- I : graph $G=(V,E)$
- S : $\{U \subseteq V \mid \{u,v\} \in E \ \ \forall u,v \in U\}$
- m(G,U) = $|U|$
- goal = *max*

We can describe optimization problems in the following more informal and simpler form:

Max Clique

- INPUT: graph $G=(V,E)$
- SOLUTION: $U \subseteq V$ such that $\{u,v\} \in E \ \forall u,v \in U$
- MEASURE: $|U|$

Other problems …

Min Vertex Cover

- INPUT: graph $G=(V,E)$
- SOLUTION: $U \subseteq V$ such that $u \in U$ or $v \in U$ $\forall \{u,v\} \in E$
- MEASURE: $|U|$

Min TSP (Traveling Salesman Problem)

- INPUT: set of cities $C=\{c_1, c_2, \dots, c_n\}$ and distances $d(c_i,c_j)\in N$ for every pair of cities $c_i,c_j \in C^2$
- SOLUTION: A tour of all the cities, that is a permutation $<c_{p(1)}, c_{p(2)}, \dots, c_{p(n)}>$ describing the order of visit of the cities
- MEASURE: tour length, that is

$$\left(\sum_{i=1}^{n-1} d(c_{p(i)}, c_{p(i+1)})\right) + d(c_{p(n)}, c_{p(1)})$$

Optimal solution

Def: Given an instance $x \in I$ of π, a solution $y^* \in S(x)$ is optimal for x if $m(x,y^*) = goal\ \{m(x,y) | y \in S(x)\}$.

The measure of an optimal solution (or analogously of all the optimal solutions) of x is denoted as $m^*(x)$ or simply m^*.

Underlying decision problem

Every optimization problem has an underlying decision problem that can be obtained by introducing an integer k to the input instance and asking whether there exists a feasible solution of measure $\leq k$ (for MIN) and $\geq k$ (for MAX).

- Optimization problem: given an input x, find $y \in S(x)$ such that $m(x,y)$ is minimum or maximum (according to the goal)
- Underlying decision problem: given an input x and an integer $k \geq 0$, is there $y \in S(x)$ such that $m(x,y) \leq k$ (MIN) or $m(x,y) \geq k$ (MAX)?

Example: underlying decision problem of Max Clique

Max Clique

- INPUT: $G=(V,E)$
- SOLUTION: $U \subseteq V$ such that $\{u,v\} \in E \ \ \forall u,v \in U$
- MEASURE: $|U|$

Underlying decision problem:

- INPUT: $G=(V,E)$ and integer $k>0$
- QUESTION : is there a clique U in G such that $|U| \geq k$?

Remarks

- If there exists a polynomial algorithm *A* for the optimization problem then there exists a polynomial algorithm also for the underlying decision problem that works as follows:

  1. Executes *A* for determining the optimum solution y^* for input x
  2. Answers *1* if $m(x,y^*)\leq k$ (MIN) or $m(x,y^*)\geq k$ (MAX)

$$\Downarrow$$

- The optimization problem is at least as difficult as the underlying decision problem.

Complexity classes of optimization problems: PO

- An optimization problem π belongs to class *PO* if:
  - for every input x, $x \in I$ can be checked in polynomial time,
  - there exists a polynomial p such that for every $x \in I$ and $y \in S(x)$ it is $|y| \leq p(|x|)$,
  - for every $x \in I$ and $y \in S(x)$, $m(x,y)$ can be computed in polynomial time (with respect to $|x|$),
  - for every $x \in I$, an optimal solution y^* can be computed in polynomial time.

Ex.: Shortest path between two nodes, minimum spanning tree, etc…

Complexity classes of optimization problems: NPO

- An optimization problem π belongs to class *NPO* if:

  - for every input x, $x \in I$ can be checked in polynomial time,
  - there exists a polynomial p such that for every $x \in I$ and $y \in S(x)$ it is $|y| \leq p(|x|)$,
  - for every $x \in I$ and $y \in S(x)$, $m(x,y)$ can be computed in polynomial time (with respect to $|x|$),
  - ~~for every $x \in I$, an optimal solution y^* can be computed in polynomial time.~~

Ex.: Max Clique, Min Vertex Cover, Min TSP, ecc…

In practice …

- *PO* = class optimization problems whose underlying decision problem belongs to *P*

- *NPO* = class optimization problems whose underlying decision problem belongs to *NP*

Clearly $PO \subseteq NPO$

- Def. An optimization problem in *NPO* is *NP-HARD* if its underlying decision problem is *NP-Complete*

- Theorem: If *P≠NP* a *NP-HARD* optimization problem cannot be solved in polynomial time (since it is at least as difficult as the underlying decision problem)

- Theorem: If *P=NP* then *PO=NPO*

Almost all the problems we will see in the sequel are *NP-HARD*, that is not efficiently solvable.

We will design for such problems algorithms that return solutions "close" to optimal ones.