Tutorial Exact Algorithms

Exercise T27 Single Machine Scheduling. We are given a set of n tasks $T_1, \ldots, T_n$ and each task has associated with it three integers: a positive execution time $l(i)$; a non-negative release time $r(i)$; and, a positive deadline $d(i)$. We wish to determine whether there exists a single-machine schedule starting at time 0 such that the execution of each task takes place in the interval bounded by its release time and its deadline. A feasible solution may be viewed as a set $\mathscr{I}$ of disjoint open intervals on the real line such that the following hold:

1. Each interval I is assigned a label from the set $\{1, \ldots, n\}$ and if I is assigned label i, then $I = (a, a + l(i))$, where a is an integer satisfying $r(i) \leq a$ and $a + l(i) \leq d(i)$.
2. Each integer from $\{1, \ldots, n\}$ is a label of some $I \in \mathscr{I}$.

Use Inclusion-Exclusion to find out the number of feasible schedules.

Solution We will use the framework of Inclusion-Exclusion to count the number of feasible solutions. A feasible solution may be viewed as a set $\mathscr{I}$ of disjoint open intervals on the real line such that the following hold:

1. Each interval I is assigned a label from the set $\{1, \ldots, n\}$ and if I is assigned label i, then $I = (a, a + l(i))$, where a is an integer satisfying $r(i) \leq a$ and $a + l(i) \leq d(i)$.
2. Each integer from $\{1, \ldots, n\}$ is a label of some $I \in \mathscr{I}$.

To use the framework on Inclusion-Exclusion, we must define what the objects are and what properties they must satisfy. The objects in this setting are collections of disjoint intervals satisfying condition (1) above. An object $\mathscr{I}$ satisfies property A_i for $i \in [n]$ if there exists an $I \in \mathscr{I}$ with label i. Therefore an object $\mathscr{I}$ that satisfies properties A_i for all $1 \leq i \leq n$ is a feasible solution. By IE,

$$|A_1 \cap \cdots \cap A_n| = \sum_{X \subseteq [n]} (-1)^{|X|} \left| \bigcap_{i \in X} \bar{A}_i \right|.$$

Now $\left|\bigcap_{i \in X} \bar{A}_i\right|$ is the number of collections of disjoint open intervals that satisfy (1) above but do not contain labels from X.
We will compute $\left|\bigcap_{i \in X} \bar{A}_i\right|$ using dynamic programming. Let $T = \max_i d(i)$ and for $0 \leq t \leq T$, define

$$E(t) := \{i \in [n] \colon r(i) \leq t - l(i) \text{ and } t \leq d(i)\}.$$

That is, $E(t)$ is the set of tasks that are eligible to be completed within time t. Let $P_X(t)$ be the number of orderings of disjoint open intervals such that (1) is satisfied and no interval is labeled by an element of X. Then $\left|\bigcap_{i \in X} \bar{A}_i\right| = P_X(T)$. Now $P_X(0) = 1$, since the empty collection of intervals vacuously satisfies (1) and has no labels from X. Also define $P_X(t) = 0$ for all $t < 0$. For $t \geq 1$

$$P_X(t) = P_X(t-1) + \sum_{i \in E(t) \setminus X} P_X(t - l(i)).$$

The time taken to compute $P_X(T)$ is $O(nT)$ and the space required is $O(\max_i l(i))$ and hence the number of feasible solutions can be counted in time $O^*(2^n)$ and space $O^*(\max_i l(i))$.

Exercise T28 MAX INTERNAL SPANNING TREE. Let T be a tree with at least three vertices. A vertex $v \in V(T)$ is a leaf if $\deg(v) \leq 1$. A vertex is *internal* if it is not a leaf. If the tree is rooted at r, then the root is defined *neither* an internal vertex nor a leaf. The MAX INTERNAL SPANNING TREE problem is defined as follows:

Input:	A connected graph $G = (V, E)$ and a positive integer $1 \leq c \leq \lvert V \rvert$.
Question:	Is there a spanning tree of G with at least c internal vertices?

The objective of this exercise and Assignment H29 is to solve this problem using Inclusion-Exclusion.

1. We first establish a connection with branching walks. Show that: *There exists a spanning tree of G with at least c internal vertices iff there exists a vertex $s \in V$ and a branching walk $B = (T, \varphi)$ from s of length $|V| - 1$ such that $\varphi(V(T)) = V$ and T has at most $|V| - (c+1)$ leaves.* Assume that $|V| \geq 3$.

2. We now count the number of branching walks with the properties in (1) above using Inclusion-Exclusion. To do this we need to define what the objects are and what properties they must satisfy. The objects in this setting are all branching walks $B = (T, \varphi)$ of length $n - 1$ such that T has at most $n - (c+1)$ leaves. An object $B = (T, \varphi)$ has property A_v for $v \in V$ if $v \in \varphi(V(T))$. Formulate the problem of counting branching walks as an IE-expression.

3. For $X \subseteq V$, what does the term $\left|\bigcap_{u \in X} \bar{A}_i\right|$ represent? We will now evaluate this term. For $X \subseteq V$, $u \in V \setminus X$, $1 \leq l \leq n-1$ and $1 \leq q \leq n-(c+1)$, define $M_X(u, l, q)$ to be the set of all branching walks in $G[V \setminus X]$ from u of length l and with q leaves. Also define $m_X(u, l, q) := |M_X(u, l, q)|$. Write down $\left|\bigcap_{u \in X} \bar{A}_i\right|$ in terms of $M_X(u, l, q)$.

4. What is $m_X(u, 0, q)$? For $l \geq 1$, write down a recurrence for $m_X(u, l, q)$.

Solution The connection with branching walks may be proved as follows:

Lemma 1 *There exists a spanning tree of G with at least c internal vertices iff there exists a vertex $s \in V$ and a branching walk $B = (T, \varphi)$ from s of length $|V| - 1$ such that $\varphi(V(T)) = V$ and T has at most $|V| - (c+1)$ leaves.*

Proof. For the forward direction, let $S = (V, T')$ be a spanning tree of G with at least c internal vertices. Root S at a leaf node s and arbitrarily order the vertices of S. Since the root is neither a leaf nor an internal vertex, the number of leaves in the rooted ordered tree S is at most $n - (c+1)$. Define $\varphi \colon V(S) \to V(S)$ to be the identity map. Clearly (S, φ) is a branching walk of G with the stated properties.

For the reverse direction, let $B = (T, \varphi)$ be a branching walk such that $\varphi(V(T)) = V$ and T has at most $|V| - (c+1)$ leaves. Note that φ is a one-one map in this case and that $(\varphi(V(T)), \varphi(E(T)))$ is actually a spanning tree of G with at most $n - c$ leaves. The spanning tree $(\varphi(V(T)), \varphi(E(T)))$ might have one leaf more than T because the root of T might be of degree one. Hence the spanning tree $(\varphi(V(T)), \varphi(E(T)))$ has at least c internal vertices. ■

Now the objects in this setting are all branching walks $B = (T, \varphi)$ of length $n - 1$ such that T has at most $n - (c+1)$ leaves. An object $B = (T, \varphi)$ has property A_v for $v \in V$ if $v \in \varphi(V(T))$. Therefore an object that satisfies properties A_v for all $v \in V$ represents a spanning tree with at least c internal vertices. By IE,

$$|A_1 \cap \cdots \cap A_n| = \sum_{X \subseteq V} (-1)^{|X|} \left| \bigcap_{u \in X} \bar{A}_i \right|.$$

Now the term $\left|\bigcap_{u \in X} \bar{A}_i\right|$ represents the number of branching walks $B = (T, \varphi)$ of length $n - 1$ such that T has at most $n - (c+1)$ leaves and $X \cap \varphi(V(T)) = \emptyset$.
For $X \subseteq V$, $u \in V \setminus X$, $1 \le l \le n - 1$ and $1 \le q \le n - (c+1)$, define $M_X(u, l, q)$ to be the set of all branching walks in $G[V \setminus X]$ from u of length l and with q leaves. Also define $m_X(u, l, q) := |M_X(u, l, q)|$. Note that

$$\left| \bigcap_{i \in X} \bar{A}_i \right| = \sum_{u \in V \setminus X} m_X(u, n-1, n-(c+1)).$$

Now

$$m_X(u, 0, q) = \begin{cases} 1 & \text{if } q = 1. \\ 0 & \text{if } q \ge 2. \end{cases}$$

For $l \ge 1$,

$$m_X(u, l, q) = \sum_{v \in N(u) \setminus X} \sum_{l_1 + l_2 = l-1} \sum_{q_1 + q_2 = q} m_x(u, l_1, q_1) \cdot m_x(v, l_2, q_2).$$

To prove the above, it is sufficient to show that the map

$$\gamma \colon \bigcup_{v \in N(u) \setminus X} \bigcup_{l_1, l_2} \bigcup_{q_1, q_2} M_X(u, l_1, q_1) \times M_X(v, l_2, q_2) \to M_X(u, l_1 + l_2 + 1, q_1 + q_2)$$

defined by

$$\gamma((T_1, \varphi_1), (T_2, \varphi_2)) \mapsto (T_1 \bullet T_2, \varphi_1 \cup \varphi_2)$$

is a bijection (where $T_1 \bullet T_2$ and $\varphi_1 \cup \varphi_2$ is as defined in class).

Homework Assignment H28 (10 Points) In the context of the Single Machine Scheduling problem, the total completion time of a schedule $T_{\pi(1)}, \ldots, T_{\pi(n)}$ is defined as time at which the last job $T_{\pi(n)}$ finishes. Use Inclusion-Exclusion to obtain the minimum completion time of a feasible schedule.

Homework Assignment H29 (10 Points) Complete the solution of the Max Internal Spanning Tree problem by proving that the recurrence for $m_X(u, l, q)$ is correct. Analyze the time and space complexity of evaluating $m_X(u, l, q)$ for a fixed $X \subseteq V$ and all u, l, q. What is the time and space complexity of the algorithm deciding Max Internal Spanning Tree?