ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Mohsen Ghaffari, Angelika Steger, David Steurer, Emo Welzl, Peter Widmayer

Solution 1

(a) We prove that the optimal value of the given integer linear programming is equal to the minimum cost of a tree which contains the root r.

Assume that there is a tree T whose cost is equal to c and includes r. We prove that there is a feasible solution to the given integer linear programming for which the objective function is equal to c. Set $x_e = 1$ if edge $e \in E(T)$ and $x_e = 0$ otherwise. Furthermore, set $y_v = 1$ if $v \in V(T)$, and $y_v = 0$ otherwise; this immediately implies that $y_r = 1$. Consider an arbitrary non-empty set $S \subseteq V \setminus \{r\}$ and some node $v \in S$. If $y_v = 0$, then clearly $\sum_{e\in\delta(S)} x_e \geq y_v = 0$ because $x_e \in \{0,1\}$. Otherwise, $y_v = 1$ which implies that $v \in V(T)$. Thus, there is a path from v to r. In other words, there is an edge $e \in \delta(S)$ such that $e \in E(T)$, that is $x_e = 1$. Thus, $\sum_{e\in\delta(S)} x_e \geq y_v$. Overall, all constraints are satisfied. Furthermore, it is easy to see that the objective function in this setting is equal to c, the cost of tree T.

Now, assume that we have a feasible point $(\mathbf{x},\mathbf{y})$ of the integer linear program for which the objective function is equal to c. We will prove that there is a tree with cost at most c which contains r. Consider first some vertex $v \in V$ with $y_v = 1$. We claim that there is a path from v to r in G along which every edge e has $x_e = 1$. To show this, consider first the set $S = \{v\}$ and assume that $v \neq r$ as otherwise there is nothing to show. By the constraints

$$\sum_{e\in\delta(S)} x_e \geq y_v \quad \forall S \subseteq V \setminus \{r\}, S \neq \emptyset, \forall v \in S$$

there is an edge from v to another node u in V. If u is equal to r, we are done. If not, repeat the same argument on the set $S = \{v,u\}$. The claim follows by induction on $|S|$. For every $v \in V$ with $y_v = 1$ fix such a path $P_v = (V_v, E_v)$ from v to r where $V_v \subseteq V$ are the vertices on the path and $E_v \subseteq E$ are the edges on the path and define T to be some spanning tree of the graph

$$G' = \left(\bigcup_{v\in V, y_v=1} V_v, \bigcup_{v\in V, y_v=1} E_v \right).$$

In other words, T is some spanning tree on the graph consisting of the union of all the paths P_v, where $y_v = 1$. We claim that the cost of T is at most c. By definition T contains all vertices $v \in V$ with $y_v = 1$, and potentially some others, so the vertex penalty cost of T is less than the corresponding cost in the feasible solution. On the other hand, all edges contained in T have $x_e = 1$ in the feasible solution of the LP, so the cost of T is at most c.

(b) We relax the integer linear programming formulation by replacing the constraints $y_v \in \{0,1\}$ and $x_e \in \{0,1\}$ with $0 \leq y_v \leq 1$ and $0 \leq x_e \leq 1$, respectively. We can then use the ellipsoid method to solve the resulting linear program in polynomial time in the size of the input. As seen in the lecture, all we need to provide is a separation oracle; that is, an efficient algorithm that takes a vector $(\mathbf{x},\mathbf{y})$ as input and returns YES if the given vector satisfies all constraints or, otherwise, returns NO and one specific inequality that is violated by the vector.

Since there is an exponential number of them and we do not have enough time to go through all of them one by one, the only non-trivial part is to check whether the constraints

$$\sum_{e\in\delta(S)} x_e \geq y_v \qquad\qquad \forall S \subseteq V \setminus \{r\}, S \neq \emptyset, \forall v \in S$$

are satisfied for some given $(\mathbf{x}, \mathbf{y})$. For this purpose, we construct a network flow problem on the graph $G = (V, E)$ (as given in the exercise) in which the capacity of each edge $e \in E$ is set to x_e. For every vertex $v \in V \setminus \{r\}$, we then check whether the maximum flow from v to the root r is at least y_v by using any polynomial-time flow algorithm.

- If the maximum flow is strictly less than y_v, then by the maxflow-mincut theorem there must be a cut S that contains v but not r with capacity strictly less than y_v, and which can also be found in polynomial time. Such a cut S combined with the vertex v provides us with the desired violated inequality $\sum_{e\in\delta(S)} x_e \not\geq y_v$. Hence, if we land in this first case for *any* $v \in V \setminus \{r\}$, we can return NO and the corresponding violated inequality.
- If the maximum flow is at least y_v, then for all cuts S that contain v but not r we clearly must have $\sum_{e\in\delta(S)} x_e \geq y_v$, for otherwise the flow could not be as large as it is. Hence, if for *every* $v \in V \setminus \{r\}$ we land in this second case, we can safely return YES.

(c) $V \setminus V(T) \subseteq V$ and $y_v^* \leq 1$ gives us

$$\frac{1}{1-\alpha}\sum_{v\in V} \pi_v(1-y_v^*) \geq \frac{1}{1-\alpha}\sum_{v\in V\setminus V(T)} \pi_v(1-y_v^*).$$

By observing $y_v^* < \alpha \; \forall v \in V \setminus V(T)$ we can bound $(1-y_v^*) > (1-\alpha)$ or equivalently $\frac{1-y_v^*}{1-\alpha} > 1$ which gives us

$$\frac{1}{1-\alpha}\sum_{v\in V} \pi_v(1-y_v^*) \geq \frac{1}{1-\alpha}\sum_{v\in V\setminus V(T)} \pi_v(1-y_v^*) > \sum_{v\in V\setminus V(T)} \pi_v$$

(d) Let μ be the cost of the optimal solution to this problem. In (a) we showed that an optimal solution of the integer program has cost μ as well. In (b) we showed that we can calculate an optimal solution (x^*, y^*) to the relaxed LP in polynomial time. Given a feasible solution (x, y) to our LP let $c(x, y)$ be the cost that we're trying to minimize. Clearly $c(x^*, y^*) \leq \mu$ holds since the integer solutions are a subset of the LP solutions.

Now if we assume that we can find the tree T from part (c) in polynomial time and that the additional inequality holds as well we show that T is a 3-approximation of the optimal solution if we choose a suitable $\alpha > 0$. By combining the two inequalities from (c) and (d) we get

$$\sum_{e\in E(T)} c_e + \sum_{v\in V\setminus V(T)} \pi_v \leq \frac{2}{\alpha}\sum_{e\in E} c_e x_e^* + \frac{1}{1-\alpha}\sum_{v\in V} \pi_v(1-y_v^*).$$

Note that we have the cost of our solution T on the left hand side. We can bound the right hand side in the following way:

$$\sum_{e\in E(T)} c_e + \sum_{v\in V\setminus V(T)} \pi_v \leq \max(\frac{2}{\alpha}, \frac{1}{1-\alpha})(\sum_{e\in E} c_e x_e^* + \sum_{v\in V} \pi_v(1-y_v^*))$$

This maximum can be minimized by setting $\frac{2}{\alpha} = \frac{1}{1-\alpha}$; thus, we obtain $\alpha = \frac{2}{3}$. Setting $\alpha = \frac{2}{3}$ in the previous inequality gives us

$$\sum_{e\in E(T)} c_e + \sum_{v\in V\setminus V(T)} \pi_v \leq 3(\sum_{e\in E} c_e x_e^* + \sum_{v\in V} \pi_v(1-y_v^*)) = 3c(x^*, y^*) \leq 3\mu.$$

Thus, our solution T will not be more than a factor 3 times the value of the minimum solution. Note that it takes polynomial time to get T because it takes polynomial time to get the optimal solution $c(x^*, y^*)$ for the relaxed LP. Once we have $c(x^*, y^*)$ by assumption we can generate T in polynomial time.

Solution 2

(a) Let the non-negative variable x_i for $1 \le i \le n$ denote the produced amount of product i. We want to maximize $\sum_{i=1}^{n} x_i p_i$, which is our final profit. However, we need $\sum_{i=1}^{n} x_i b_{ij} \le a_j$ for $1 \le j \le m$ because for each material of kind j we only have j units available. Therefore, the following linear program models our problem.

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} x_i p_i \\ & \\ \text{subject to} & \sum_{i=1}^{n} x_i b_{ij} \le a_j \quad \forall 1 \le j \le m, \\ & \qquad x_i \ge 0 \quad \forall 1 \le i \le n. \end{array}$$

It is dual is equal to

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{m} y_j a_j \\ & \\ \text{subject to} & \sum_{j=1}^{m} y_j b_{ij} \ge p_i \quad \forall 1 \le i \le n, \\ & \qquad y_j \ge 0 \quad \forall 1 \le j \le m. \end{array}$$

(b) **"(1) ⇒ (2)"**: Assume that (1) holds. First we note that

$$c^T x^* \overset{(i)}{\le} (A^T y^*)^T x^* = y^T A x \overset{(ii)}{\le} (y^*)^T b = b^T y^*. \tag{$*$}$$

Here (i) follows from $x^* \ge 0$, $A^T y^* \ge c$. Similarly, (ii) follows from $y^* \ge 0$, $Ax^* \le b$.

By strong duality, $c^T x^* = b^T y^*$. Hence both inequalities in $(*)$ are actually equalities. Consider the first of them, which we now write as in

$$0 = (A^T y - c)^T x = \sum_{i=1}^{n} \underbrace{\left(A^T y - c\right)_i}_{\ge 0} \underbrace{x_i}_{\ge 0}.$$

Since every term in the sum is non-negative, all the terms in the sum must vanish. That is, we have $\left(A^T y - c\right)_i x_i = 0$ for all $i = 1, \dots, n$. Equivalently,

$$\left(A^T y\right)_i = c_i \quad \text{or} \quad x_i = 0, \qquad \text{for all } i = 1, \dots, n.$$

This shows one part of (2). The proof of the other part of (2) is analogous.

"(2) ⇒ (1)": Assume that (2) holds, that is, we have

$$\begin{array}{rl} \text{for all } i: & \left(A^T y\right)_i = c_i \quad \text{or} \quad x_i = 0; \\ \text{for all } j: & (Ax)_j = b_j \quad \text{or} \quad y_j = 0. \end{array}$$

Summing up all these equations, we obtain

$$0 = (A^T y^* - c)^T x^* + (b - Ax^*)^T y = b^T y^* - c^T x^*,$$

hence $c^T x^* = b^T y^*$. From weak duality we know that, for any feasible solution x, the term $c^T x$ cannot be larger than $b^T y^*$. Hence $c^T x^*$ is the optimum of (P), and x^* is an optimal solution for (P). For similar reasons, y^* is an optimal solution for (D).

Solution 3

(a) Assume that Hansel and Gretel have the n-bit binary numbers $a_0, \cdots, a_{n-1}$ and $b_0, \cdots, b_{n-1}$, respectively. Define $a := \sum_{i=0}^{n-1} a_i 2^i$ and $b := \sum_{i=0}^{n-1} b_i 2^i$. Furthermore, let $f_p(x) = x \mod p$ for an integer x and prime p. We ask Hansel to pick a prime number p between 1 and n^4 uniformly at random and then send p and $f_p(a)$ to Gretel. Then, Gretel will compute $f_p(b)$. If $f_P(a) = f_p(b)$, she decides that a and b are equivalent and they are not otherwise.

Since $p \leq n^4$ and $f_p(a) < p$, Hansel needs $\mathcal{O}(\log n)$ bits to send both p and $f_p(a)$. Furthermore, in case of $a = b$, Gretel will get the right answer because $f_p(a) = f_p(b)$, regardless of the choice of p. If $f_p(a) \neq f_p(b)$, she concludes that $a \neq b$, which is correct. Thus, the only possible scenario which might result in a wrong decision is that $a \neq b$ but $f_p(a) = f_p(b)$, that is $f_p(c) = 0$ for $c = |a - b|$. Based on Proposition 2, there are $\Omega(\frac{n^4}{\log n^4})$ distinct prime numbers smaller than n^4. Furthermore by Proposition 1, the number of prime divisors of c is at most n (note that $c < 2^n$). Thus, the probability that p divides c is upper-bounded by $\frac{n}{\Omega(n^4/\log n^4)}$, which is clearly smaller than $\mathcal{O}(\frac{1}{n^2})$. Thus, she makes the right decision with probability $1 - \mathcal{O}(\frac{1}{n^2})$.

(b) Let $G = (V, E)$ be a graph with at least one Pfaffian orientation (otherwise G has 0 Pfaffian orientations and the claim holds) and let $\overrightarrow{H}$ be some fixed Pfaffian orientation of G. We Represent an orientation $\overrightarrow{G}$ of the graph G by a vector $v(\overrightarrow{G}) \in GF(2)^E$ that is 0 at position e if and only if it is oriented in the same direction in both $\overrightarrow{G}$ and $\overrightarrow{H}$.

By the lecture notes an orientation of a graph is Pfaffian if and only if every nice cycle in the graph is oddly oriented. Therefore C is oddly oriented in $\overrightarrow{G}$ if and only if an even number of edges on C have an orientation that is different from the one in $\overrightarrow{H}$. In terms of the vector $v(\overrightarrow{G})$ this means that among the entries corresponding to edges appearing on C there are an even number that are set to 1. We can express such a condition as a linear constraint over $GF(2)$. Therefore the Pfaffian orientations of G are in bijection with solutions of a linear system consisting of all such constraints and we know that the number of solutions to such a system is a power of 2 (by the initial assumption there is at least one solution).