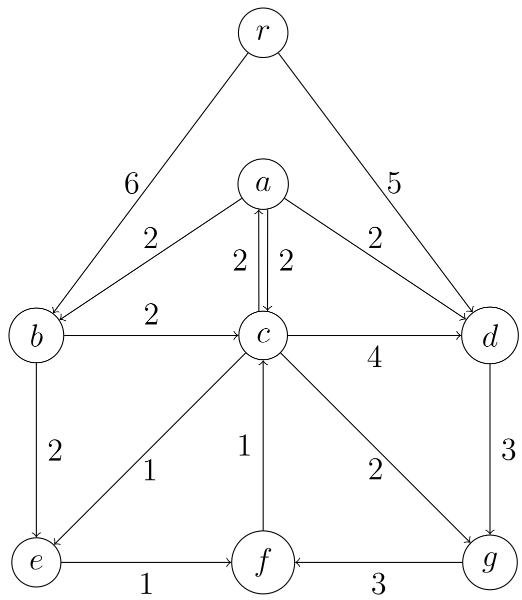
Combinatorial Optimization

Adrian Bock Fall 2011

Sheet 6 December 1, 2011

General remark:
In order to obtain a bonus for the final grading, you may hand in written solutions to the exercises marked with a star at the beginning of the exercise session on December 13.

Exercise 1
Trace the steps of algorithm from the lecture to compute a minimum weight arborescence rooted at r in the following example.



Prove the optimality of your solution!

Solution

Follow algorithm for min cost arborescence.
We find an r-arborescence of weight 16. The following 16 r-cuts prove the optimality of this solution:
$\{a\}$, $\{b\}$, $\{d\}$, $\{g\}$, $\{a,b,d\}$ two times each and $\{c\}$, $\{e\}$, $\{f\}$, $\{c,e,f\}$, $\{a,b,c,e,f,g\}$, $\{a,b,c,d,e,f,g\}$ once.

Exercise 2

Let $G = (A \cup B, E)$ be a bipartite graph. We define two partition matroids $M_1 = (E, \mathcal{I}_1)$ and $M_2 = (E, \mathcal{I}_2)$ with $\mathcal{I}_1 = \{I \subseteq E \colon |I \cap \delta(a)| \leq 1 \quad \text{for all } a \in A\}$ and $\mathcal{I}_2 = \{I \subseteq E \colon |I \cap \delta(b)| \leq 1 \quad \text{for all } b \in B\}$. What is a set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ in terms of graph theory? Can you maximize a weight function $w : E \to \mathbb{R}$ over the intersection?

Remark: This is a special case of the optimization over the intersection of two matroids. It can be shown that all such matroid intersection problems can be solved efficiently.

Solution

It is a bipartite matching. We can even find a maximum weight matching in a general graph (ignoring the matroid structure).

Exercise 3 ($\star$)

Recall that a digraph $D = (V, A)$ is called a branching in D if the underlying undirected graph is a forest and each vertex $v \in V$ has at most one incoming arc.

(i) Let $D = (V, A)$ be a digraph and let $\mathcal{B}$ be a set of all branchings in D (i.e., subsets $B \subseteq A$ such that (V, B) is a branching). Show that $(A, \mathcal{B})$ is an intersection of two matroids.
(ii) Let $r \in V$. Show how to model the arborescences rooted at r using the intersection of two matroids.

Solution

(i) intersection of forest matroid and partition matroid.
(ii) add condition $|\delta_I^{in}(r)| = 0$ to the partition matroid (remains matroid!) and consider the intersection with the forest matroid. The elements of maximum cardinality are the r-arborescences.

Exercise 4

Describe a linear time algorithm which for any instance of the Satisfiability problem finds a truth assignment that satisfies at least half of the clauses.

Solution

Choose any truth assignment. Check whether it fulfills at least half of the clauses. If not, take the negated truth assignment (every variable previously set to true is set to false and vice versa). Clearly, this truth assignment must verify at least half of the clauses.

Exercise 5 ($\star$)

Show that deciding if a polyhedron contains an integer point is NP-complete. To do so, consider the following problem:

INTEGER LINEAR INEQUALITIES
Given: a matrix $A \in \mathbb{Z}^{m \times n}$ and a vector $b \in \mathbb{Z}^m$
Task: Is there a vector $x \in \mathbb{Z}^n$ such that $Ax \leq b$?

Show that

(i) Integer Linear Inequalities is in NP.
(ii) Integer Linear Inequalities is NP-complete.

Solution

A feasible solution is the certificate. Since matrix multiplication and comparing two vectors componentwise can be done in polynomial time in the input length, ILI is in NP.
Idea for the reduction from 3-SAT:
For clauses $C_1, \ldots, C_m$ with variables $x_1, \ldots, x_n$, we consider variables $x'_1, \ldots, x'_n$ with $0 \leq x'_i \leq 1$. For each clause C_j, we construct a constraint in which each appearance of a negated variable $\bar{x}_i$ is represented by $1 - x'_i$ and each appearance of a non-negated variable x_i is represented by x'_i.
Example: $C_j = (x_2 \vee \bar{x}_3 \vee x_4)$ correspondes to the constraint $x'_2 + (1 - x'_3) + x'_4 \geq 1$.
Clearly every satisfying truth assignment corresponds to a feasible integral solution:
We set the variable $x'_i = 1$ if x_i is set to True and $x'_i = 0$ otherwise. Since every clause is satisfied, there is at least one term in every constraint that evaluates to 1.
The converse holds as well: We set the variable x_i to True whenever $x'_i = 1$ and to False otherwise. Since every constraint is fulfilled, there is at least one variable in each clause that is set to True.

Exercise 6

Consider the following problem:

DOMINATING SET
Given: an undirected graph $G = (V, E)$ and a number $k \in \mathbb{N}$
Task: Is there a set $X \subseteq V$ with $\lvert X \rvert \leq k$ and for every $v \in V \setminus X$, we have $\{x, v\} \in E$?

Show that

(i) Dominating Set is in NP.
(ii) Dominating Set is NP-complete.

Hint: Vertex Cover

Solution

Clearly Dominating Set is in NP. Given a dominating set, one can verify in polynomial time if that is a dominating set. This can be done by taking each vertex and checking if it is either in the given set or one of its edges travel into the set. To show that is NP-complete, first of all notice that a dominating set has to include all isolated vertices (those which have no edges from them). So let us assume that our graph does not have any isolated vertices. We will show that Dominating Set is NP-complete using a reduction from Vertex Cover. Given a graph G, we will construct a graph G' as follows. G' has all edges and vertices of G. Also, for every edge $u, v \in G$, we add intermediate node on a parallel path in G' . Keeping u, v intact in G' , we add vertex w and edges u, w and w, v in G' . Now we will show that G has a vertex cover of size k if and only if G' has a dominating set of the same size. If S is a vertex cover in G, we will show that S is a dominating set for G' . S is a vertex cover, this means that every edge in G has at least one of its end points in S . Consider $v \in G'$. If v is an original node in G, then either $v \in S$ or there must be some edge connecting v to some other vertex u. Since S is a vertex cover, v is not in S , then u must be in S , and hence there is an adjacent vertex of v in S . So v is covered by some element in S . However, if w is an additional node in G' , then w has two adjacent vertices $u, v \in G$ and using the above argument at least one of them is in S . So the additional nodes are also covered by S . So if G has a vertex cover, then G' has a dominating set of at most the same size (in fact the same set itself would do). If G' has a dominating set D of size k, then look at all the additional vertices $w \in D$. Notice that w must be connected to exactly 2 vertices $u, v \in G$. Now see that we can safely replace w by one of u or v. w in D will help us dominate only $u, v, w \in G'$. But these three edges form a 3-cycle, and we can as well pick u or v and still dominate all the vertices that w used to dominate. So we can eliminate all the additional vertices as above. Since all the additional vertices correspond to one of the edges in G, and since all of the additional vertices are covered by the modified D, this means that all the edges in G are covered by the set. So if G' has a dominating set of size k, then G has a vertex cover of size at most k.

So we have proved both sides of equivalence. A dominating set of size k exists in G' if and only if a vertex cover of size k exists in G. Since we know that vertex cover is an NP-complete problem, Dominating Set is also NP-complete.