# Mastermath & Lnmb

Written exam for the Mastermath course on Discrete Optimization

December 20, 2010, 13.15 — 16.15

**Note: The exercises must be handed in, together with your solutions**

In solving the exercises, please be clear, precise and brief in your statements and conclusions. No prosa. Even if you do not solve the first part(s) of an exercise, you may use the corresponding results to solve the remaining part(s).

## Problem Collection

Bin Packing Given $n$ items with nonnegative, integral sizes $s_1, \dots, s_n$, items have to be packed into identical bins. Each bin has capacity $S$, and we assume $s_i \leq S$ for all $i = 1, \dots, n$. A *bin packing* is feasible if the bin capacity $S$ is not exceeded by the items in any bin. The *bin packing problem* asks to compute a feasible bin packing with as few bins as possible.

Clique Given an undirected graph $G = (V, E)$, a *clique* is a subset of nodes $U \subseteq V$ such that any two nodes in $U$ are adjacent. The decision problem is: Given an integer $k \leq |V|$, does there exist a clique/independent set of size at least $k$ in $G$? Both problems are known to be strongly $\mathcal{NP}$-complete.

Hamiltonian Path Given an undirected graph $G = (V, E)$, does there exist a simple path that visits each of the vertices exactly once? This problem is known to be strongly $\mathcal{NP}$-complete.

Matching Given an undirected graph $G = (V, E)$, a *matching* $M \subseteq E$ is a set of non-incident edges. The *matching* problem is to decide, for an arbitrary $k$, if a matching of size $k$ exists in $G$. Edmonds' algorithm solves this problem in polynomial time.

Partition Given are $n$ integral, non-negative numbers $a_1, \dots, a_n$ with $\sum_{j=1}^{n} a_j = 2B$. The decision problem is to decide if there is a subset $W \subseteq \{1, \dots, n\}$ such that $\sum_{j \in W} a_j = B$. This problem is $\mathcal{NP}$-complete but has a pseudo-polynomial time algorithm.

Satisfiability Given $n$ Boolean variables $x_1, \dots, x_n$, and a formula $F$ that consists of the conjunction of $m$ clauses $C_i$, $F = \bigwedge_{i=1}^{m} C_i$. Each clause consists of the disjunction of some of the variables $x_j$ (or their negation $\bar{x}_j$), for example $C_5 = (x_1 \vee x_4 \vee \bar{x}_7)$. The decision problem is: Does there exist a truth assignment $x \in \{\text{false,true}\}^n$ such that $F =$true? This problem is known to be strongly $\mathcal{NP}$-complete.

Vertex Cover Given is an undirected graph $G = (V, E)$, with non-negative, integer node weights $c_v, v \in V$. A *vertex cover* is a subset $U$ of the vertices $V$ such that for any edge $e = \{v, w\} \in E$, at least one of the nodes $v$ or $w$ is in $U$. The *minimum weight vertex cover problem* asks for a vertex cover $U \subseteq V$ such that $\sum_{v \in U} c_v$ is minimized. This problem is known to be (strongly) $\mathcal{NP}$-complete.

## Questions

1. (5 points) Consider an edge-weighted graph $G = (V, E, w)$ with $w_e \geq 0$ for all $e \in E$. Let $e^*$ be the lightest edges of $G$, i.e., $w_{e^*} < w_e$ for all $e \in E$. Prove or give a counterexample for the following claims.

   (a) Edge $e^*$ is contained in all minimum spanning trees of $G$.

   (b) Edge $e^*$ is contained in all shortest path trees of $G$. [1]

2. (10 points) Which of the following is true or false, assuming $\mathcal{P} \neq \mathcal{NP}$. Please explain your answer briefly, but precisely.

   (a) There is a polynomial transformation from SATISFIABILITY to any problem in $\mathcal{NP}$.

   (b) If there is a strongly polynomial time algorithm to solve the PARTITION problem, then there is a polynomial time algorithm to solve SATISFIABILITY.

   (c) There is a polynomial transformation from MATCHING to CLIQUE.

   (d) All problems in $\mathcal{NP}$ can be reduced to each other.

   (e) The PRIMES problem "is $n$ a prime?" is in co-$\mathcal{NP}$.

3. (7 points) The following $0-1$ integer linear program models the vertex cover problem for an undirected graph $G = (V, E)$, with node weights $c_v \geq 0$, $v \in V$

$$\begin{array}{lll} \min & \sum_{v \in V} x_v c_v & \\ & x_v + x_w \geq 1 & \text{for all } \{v, w\} \in E \\ & x_v \in \{0, 1\} & \text{for all } v \in V \end{array}$$

   The LP relaxation of this 0-1-linear program is given by relaxing the integrality condition on the $0-1$ variables $x_v$:

$$\begin{array}{lll} \min & \sum_{v \in V} x_v c_v & \\ & x_v + x_w \geq 1 & \text{for all } \{v, w\} \in E \\ & x_v \geq 0 & \text{for all } v \in V \end{array}$$

   (The redundant constraints $x_v \leq 1$ for all $v \in V$ have been omitted.)

   Consider the following LP-based rounding algorithm, called algorithm $A$:

   - Solve the LP relaxation, let $x^*$ be an optimal (fractional) solution for it.
   - Define the output of $A$ as $U = \{v \in V \mid x_v^* \geq 0.5\}$.

   Show that algorithm $A$ is a 2–approximation algorithm for VERTEX COVER. (Don't forget to show that the solution computed by $A$ is actually feasible.)

[1] The shortest path tree of $G$ is the tree that is obtained as the union of all shortest paths, from some $v_0 \in V$ to all other $v \in V$

4. Given an undirected, connected graph $G = (V, E)$ with $|V| \geq 2$, the min-max degree spanning tree problem is to find a spanning tree $T$ of the graph such that the maximal degree of the nodes in $T$ is minimized. In other words, find a spanning tree $T = (V, E_T)$, $E_T \subseteq E$, that minimizes $\max_{v \in V} d_T(v)$, where $d_T(v)$ is the degree of node $v$ in $T$. For convenience, let us call this optimization problem MDST.

   (a) (2 points) Argue (briefly!) that the MDST problem has a solution with value $k = 2$ if and only if $G$ has a Hamiltonian path.

   (b) (6 points) Assuming $\mathcal{P} \neq \mathcal{NP}$, show that there cannot be an $\alpha$-approximation algorithm for the MDST optimization problem with $\alpha < \frac{3}{2}$.

5. Consider the Bin Packing problem, as defined in the problem collection. The decision version of the Bin Packing problem asks if a set of items can be packed into at most $k$ bins.

   (a) (4 points) Give an integer linear programming formulation for the decision version of the Bin Packing problem.

   (b) (5 points) Prove that this problem is $\mathcal{NP}$-hard. (You do not need to show it is in $\mathcal{NP}$.) To show $\mathcal{NP}$-hardness, transform one of the $\mathcal{NP}$-complete problems listed in the problem collection to Bin Packing.

6. (6 points) Consider a minimum cost flow problem on a network $G = (V, A)$ with arc costs $c_{ij} \geq 0$ and arc capacities $u_{ij} > 0$, $(i, j) \in A$. Let $x^*$ be a minimum cost flow, and $\pi$ be a set of corresponding node labels such that the reduced cost optimality condition is fulfilled. (That is, $c_{ij}^{\pi} = c_{ij} - \pi(i) - \pi(j) \geq 0$ for all arcs $(i, j) \in G(x^*)$.)

   Consider the residual graph $G(x^*)$, and let $G^o(x^*)$ be the graph consisting only of those arcs of $G(x^*)$ with zero reduced cost. (That is, $c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j) = 0$).

   Show that there is an alternative minimum cost flow $x'$ if and only if $G^o(x^*)$ has a directed cycle.

7. (5 bonus points) Show that there is a 2-approximation algorithm for Bin Packing.

   (Hint: A simple greedy algorithm does the job.)

Norm: 45 + 5 bonus points in total (25 points to pass) — Good luck, happy Christmas and a successful new year!