ACO Comprehensive Exam **October 5 and 6, 2009**

1. Theory of Linear Inequalities

Let $\mathbf{Z}_+$ denote the positive integers. Suppose $a, b \in \mathbf{Z}_+^n$ are vectors and $\beta \in \mathbf{Z}_+$ is a scalar . Consider a matrix M where the columns of M are exactly the integer solutions to the knapsack problem ($a^T x \leq \beta, x \geq 0$). Let $\mathbf{1}$ denote the vector of all 1's. Show that the linear programming problem

$$\min \mathbf{1}^T y \tag{1}$$
$$\text{subject to} \tag{2}$$
$$My = b, \tag{3}$$
$$y \geq 0 \tag{4}$$

has an optimal solution if and only if $a_i \leq \beta$ for $i = 1, \ldots n$.

Solution. If the condition is satisfied then the unit vectors are all solutions to the knapsack problem, so setting the corresponding components of y to b_j we get a feasible solution to the LP. Now since the LP is feasible and the objective is bounded it has an optimal solution. On the other hand, if the condition is not satisfied for some index i, then then the i-th row of matrix M is all 0's and since b_i is positive (by assumption) there can be no feasible solution to the LP.

2. Combinatorial Optimization

Consider the following Arc Covering Problem (ACP): Given a directed graph $D = (N, A)$ with node set N, arc set A, integer arc costs $c(a)$ for $a \in A$, and a subset of arcs $L \subseteq A$, find a minimum cost set of simple directed cycles covering L (a cycle cover of L). Note that the cost of a cycle is the sum of the costs of its arcs.

1. Give a polynomial time algorithm for solving the ACP. (4pts)

2. Assume that you are given a fractional optimal solution to the ACP. Show that this fractional solution can be transformed into an integral optimal solution in polynomial time. (4pts)

3. Now assume there are two cost functions, c_1 and c_2, given as input to the ACP. Define the cost of a cycle K to be $min(c_1(K), c_2(K))$. Briefly discuss if the method you provided in question 2 will transform a fractional optimal solution to the two cost function ACP to an integral optimal solution in polynomial time. (2pts)

Solution.

1. There are several ways of doing this, two easy ones are to transform the problem so that it can be solved as a flow circulation problem or a min weighted matching problem.

2. Given the fractional optimal solution, one can show that there exist augmenting cycles with 0 cost. All of these can be canceled in polynomial time to reach an optimal integral solution.

3. This is somewhat open ended. But the cycle canceling type of ideas do not work any more.

3. Analysis of Algorithms

(a) Given a complete bipartite graph $G = (U, V, U \times V)$ with non-negative weights on the edges and an integer $k > 0$, design a polynomial time algorithm to find a minimum weight matching of cardinality k. Solve this problem by reducing it to a single call of a minimum weight perfect matching algorithm.

(b) We have a graph $G = (V, E)$ with n vertices. Now suppose we would like to find a collection of matchings such that every edge of the graph is a member of (at least) one of the matchings selected. The goal is to pick the minimum number of matchings for our collection. Give an $O(\log n)$-approximation algorithm for this problem. (Hint: Consider the set cover problem).

Solution:
(a) Let $n = |U|$ and $m = |V|$. We will define a new graph $G' = (U', V', U' \times V')$ as follows. Let S contain $n - k$ vertices and let T contain $m - k$ vertices. We will let $U' = U \cup S$, $V' = V \cup T$. We give all edges in $(U \times T) \cup (V \times S)$ weight 1 and we give every edge in $S \times T$ weight 3. If we find a minimum-weight perfect matching in G', then all of the vertices in $S \cup T$ must be matched by edges of weight 1. To see this, assume that there are two vertices $s \in S$ and $t \in T$ that are matched to each other with a weight 3 edge. Then we can choose any edge (u, v) in the matching such that $u \in U$ and $v \in V$, and we could form a lighter matching by replacing (s, t) and (u, v) by weight 1 edges (s, v) and (u, t), a contradiction. Therefore all $n - k$ vertices in S are matched to vertices in V, all $m - k$ vertices in T are matched to verties in U and exactly k edges of the matching are in $U \times V$. Moreover, all ways to match $n - k$ vertices of S to V have the same weight, and similarly for the ways to match $m - k$ vertices of T to U, so the minimum weight pefect matching of G' induces a minimum weight matching of size k on G.
(b) This can be solved by reformulating the given problem as a set cover problem. Let E be our universe, and let each matching in G be a set we can include in our set cover. The approximation algorithm for set cover iteratively chooses the set that covers the maximum number of edges not yet covered. Accordingly, we start by choosing the set that covers the maximum number of edges, i.e., the maximum matching in G. We then remove all of these edges and iteratively find the maximum matching in the remaining graph until every edge in G is included in some matching. As this is just a set cover, this approximation algorithm gives an $O(\log n)$ approximation to the minimum number of matchings required.

4. Randomized Algorithms

Consider the following random variable: X is 0 or 1 with equal probability. Let M be an n by n array of independent copies of X. Thus, M takes on all 2^{n^2} boolean values equally likely. Construct a *deterministic* algorithm $A(s)$ that given a boolean string s of length $2n - 1$ outputs an n by n boolean matrix. The first, second, and third order statistics of $A(s)$ should be the same as those of M.

Thus, the algorithm $A(s)$ "simulates" a completely random boolean matrix with many fewer random bits. The simulation is only accurate up the the given statistics.

Solution. There are two approaches. The first uses Nisan's De-randomization Theorem. They need to point out that the statistics can easily be computed by a logspace machine, and so the algorithm A is easy to construct. Actually, this method will yield much better bounds, but it will definitely solve the problem.

The second uses the following idea: Pick the first row of A randomly. Then, select each row after that as the random flip of the last row based on a random bit. This uses $2n-1$ random bits. They can then show that the statistics are okay.

4. Graph Algorithms

A graph G is *k-extendable*, where $k \geq 1$ is an integer, if every matching of size at most k is a subset of a perfect matching of G. Design a polynomial-time algorithm for the following problem: Given a bipartite graph G and an integer $k \geq 1$, determine whether G is k-extendable. Prove correctness of your algorithm, and prove that it runs in polynomial time.
Hint. You may want to consider the connectivity of the directed graph obtained from G by directing all edges from A to B and contracting a perfect matching, where (A, B) is a bipartition of G.

Solution. We need an auxiliary claim. Assume for now that G is connected and that it has a perfect matching M. Let (A, B) be a bipartition of G, and let D be defined as in the hint by contracting the perfect matching M. Thus we may assume that $V(D) = M$. We claim that D is strongly k-connected if and only if G is k-extendable.

To prove the claim suppose first that G is not k-extendable. Thus G has a matching N of size at most k that does not extend to a perfect matching. Let $A_1 \subseteq A$ and $B_1 \subseteq B$ be the vertices incident with edges of N. Then $G \backslash (A_1 \cup B_1)$ has no perfect matching, and hence, by Hall's theorem, there exists a set $X \subseteq A - A_1$ such that $|N(X)| < |X| + k$, where $N(X)$ denotes the set of neighbors of X in G; that is, including neighbors in B_1. Let Z be the set of all $e \in V(D) = M$ that have an end in $N(X) - B_1$. Then $D \backslash Z$ has no edge from $\{e \in M : e \text{ has an end in } X\}$ to $\{e \in M : e \text{ has an end in } B - N(X)\}$. Since $|Z| < k$ we deduce that D is not strongly k-connected.

Conversely, suppose that D is not strongly k-connected. Then G has a matching N with $|N| < k$ such that the vertices of G not incident with edges of N can be partitioned into disjoint non-empty sets X, Y such that no edge of G has one end in $X \cap A$ and the other in $Y \cap B$. Choose such a matching with $|N|$ minimum. Suppose first that $N = \emptyset$. Then the connectivity of G implies that there exists an edge $f \in E(G)$ with one end in $X \cap B$ and the other in $Y \cap A$. But f belongs to no perfect matching of G, as desired. Thus we may assume that $N \neq \emptyset$. The minimality of N implies that D is strongly connected, and hence has a directed path from X to Y. Thus there exists an N-augmenting path in G from a vertex in $X \cap A$ to a vertex in $Y \cap B$. It follows that there exists a matching N' in G of size $|N| + 1$ that saturates all the vertices incident with N, plus a vertex in $X \cap A$ and a vertex in $Y \cap B$. But N' is a subset of no perfect matching of G, as desired.

Using the claim, a polynomial-time decision algorithm is as follows. If G is disconnected, we apply the algorithm to each component of G separately. This works, because G is k-extendable if and only if each component of G is. Thus we may assume that G is connected. If G has no perfect matching, then we answer "no" and stop. Thus we may assume that G has a perfect matching M. (This uses a standard perfect matching algorithm.) We construct the digraph D and test its strong k-connectivity, for instance by checking that every pair of distinct vertices u, v are joined by k internally disjoint directed paths from u to v.

5. Graph Theory

Let G be a 2-connected graph with vertex-set $\{1, 2, \ldots, n\}$ containing a triangle, and let $\tau_1, \tau_2, \ldots, \tau_{n-1}$ be distinct tokens. Initially, the tokens are placed on distinct vertices of the graph, at most one token per vertex. Thus exactly one vertex of G has no token on it. A *move* consists of sliding a token from its current position along an edge of G to a vertex that currently has no token on it. Prove that there exists a sequence of moves that results in token τ_i being positioned at vertex i for all $i = 1, 2, \ldots, n-1$.

Solution: We proceed by induction on n. If $n = 3$, then there are two tokens, the graph is a triangle, and so the conclusion holds. We may therefore assume that $n \geq 4$, and that the conclusion holds for all graphs on fewer than n vertices.

We notice that given any initial placement of the tokens, for every vertex $v \in V(G)$ there is a sequence of moves that makes v be the only token-free vertex. That is easy and uses only the connectivity of G. Thus it suffices to show the following:
(1) given an initial placement of the tokens, there is a sequence of moves that interchanges the positions of τ_1 and τ_2 and brings the remaining tokens back to their initial positions.

Let C be a triangle in G. To prove (1) it suffices to prove the following:
(2) given an initial placement of the tokens, there is a sequence of moves that brings τ_1 and τ_2 to C and leaves the third vertex of C token-free.

Indeed, to prove (1) we use a sequence of moves σ as in (2), then interchange the positions of τ_1 and τ_2 in C, and then we use the inverse of σ to get a token placement as required for (2).

Thus is suffices to prove (2). Since G is 2-connected and has a triangle, it has an ear-decomposition, where the first graph is C. That is, there exist subgraphs $H_0, H_1, \ldots, H_k$ such that $H_0 = C$, and for each $i = 1, 2, \ldots, k$ the graph H_i is a path with both ends in $H_0 \cup H_1 \cup \cdots \cup H_{i-1}$, and otherwise disjoint from it. Let $H = H_{k-1}$ and let $P = H_k$. If τ_1, τ_2 and the token-free vertex all belong to H, then (2) follows by induction applied to H. If only the token-free vertex fails to belong to H, then it is easy to bring it to H. So we may assume that τ_1 is not in H. By induction it is possible to move the token-free vertex to an end of P (keeping τ_2 in H, if it was already there) and subsequently move τ_1 one step closer to H. By repeating this we can move both τ_1 and τ_2 into H, and then (2) follows by induction.

6. Probability

1. Let X be a discrete time Markov chain with state space $S = \{1, 2\}$, and transition matrix

$$M = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}.$$

Classify the states of the Markov chain. Suppose that $a \cdot b > 0$ and $a \cdot b \neq 1$. Find the $n-$step transition probabilities and show directly that they converge to unique stationary distribution as $n \to \infty$.

2. Flies and wasps land on your dinner plate in the manner of independent Poisson processes with respective intensities λ and μ. Show that the arrivals of these flying insects form a Poisson process with intensity $\lambda + \mu$.

Solution: Not available.

7. Algebra

Let p, q be distinct primes. If a group G of order pq acts on a set X having $pq - p - q$ elements, show that there is a fixed point for this action (i.e., an element $x \in X$ with $g \cdot x = x$ for all $g \in G$).

Solution: Assume for the sake of contradiction that there is no fixed point in X. Let $x_1, \ldots, x_t$ be representatives for the different orbits, let $G \cdot x_i$ be the orbit of x_i, and let G_i be the stabilizer of x_i. Then $|G \cdot x_i| \leq |X| < |G|$ for all i. And as there is no fixed point in X, we have $|G \cdot x_i| \neq 1$ for all i. Since $|G \cdot x_i| = [G : G_i]$ divides $|G|$, we must have $|G \cdot x_i| \in \{p, q\}$ for all i. Since $|X| = \sum_{i=1}^{t} |G \cdot x_i|$, it follows that $pq - p - q = ap + bq$ with a, b nonnegative integers. But then p divides $b + 1$ and q divides $a + 1$, and in particular $a \geq q - 1$ and $b \geq p - 1$, so that

$$ap + bq \geq p(q-1) + q(p-1) = 2pq - p - q > pq - p - q \ ,$$

a contradiction.