CSC 2411 - Assignment 1
(Final Version)

Due Feb 5 , 2009

1. Let $U \subset \mathbb{R}^d$. Recall that conv(U), the convex hull of U, is defined as the set of all points in $\mathbb{R}^d$ that can be written as a convex combination of a (finite) set of points in U:

$$\text{conv}(U) = \left\{ \sum_{i=1}^{k} \lambda_i x_i : x_i \in U, \lambda_i \in \mathbb{R}, \lambda_i \geq 0, \sum_i \lambda_i = 1, k = 1, 2, \ldots \right\}.$$

   Show that if S is a finite subset of $\mathbb{R}^d$ of size $d+2$, it can be partitioned into two sets such that their convex hulls intersect. That is, there exists $T \subset S$, such that,

$$\text{conv}(T) \cap \text{conv}(S \backslash T) \neq \emptyset.$$

2. (a) Prove that if $x \in \mathbb{R}^d$, the following set is a convex set,

$$S = \left\{ (y, r) : y \in \mathbb{R}^d, r \in \mathbb{R}, d(x, y) \leq r \right\},$$

   where $d(x, y)$ is the usual euclidean distance, and (y, r) denotes the $d+1$ dimensional vector which is the concatenation of y and r. Notice: to show that S (which is a closed set) is convex, it is enough to show that if $z_1, z_2 \in S$ then also $(z_1 + z_2)/2 \in S$. You may use this fact without proof.

   (b) Formalize the following problem as a convex programming problem (i.e., a minimization problem with a convex feasible set). Given $x_1, \ldots, x_n \in \mathbb{R}^d$ find the smallest ball containing all of them (a ball in $\mathbb{R}^d$ centred at x and of radius r is $\{x \in \mathbb{R}^d : \|x - y\| \leq r\}$.

3. In class we've described an (exponential time) algorithm for solving LP that enumerates all BFSs and finds the best one. We have shown that this algorithm indeed works, *provided* that that the objective function is bounded. Show how to extend this algorithm so that instead of making the above assumption, detects whether the objective function is unbounded, and otherwise returns the optimum as before.

4. (a) Show that if a variable x_j leaves the basis, it cannot return to it in the next iteration of the simplex algorithm (regardless of the pivot rule) . Deduce that there are no cycles of size two in the algorithm.

(b) Give an example showing that a variable x_j can join the basis and leave it in the next iteration.

Hint: Recall that in all pivot rules, when we express the basic variable leaving the basis as an affine function of the nonbasic variables, only variables with negative coefficient ($T_{ij} > 0$ in the notation in class) may enter the basis.