Weekly Report (2009-01-10)

Hongxing Li

I. TA Work

Set up the home page for CSIS0234B and update the course information. Design the assignment 1, which is a Java project implementing a simple file transfer service.

II. Research Work

In last report, we have formulated our problem model into *Mixed Integer Linear Programming* problem, more precisely a *0-1 Mixed Integer Linear Programming* problem. After the weekly meeting, I have read some books and lecture notes to studied the *Mixed Integer Linear Programming* problem. There are mainly two solutions for this problem,

- *Branch and Bound*: Step 1: Let all variables be real numbers first and solve the optimization problem as linear programming. Step 2: Suppose the optimal assignment for the linear programming is X^* and $x_i^* = a_i.b_i$, we further discuss the optimal objective value and variable assignment for two branches: $x_i^* \leq a_i$ and $x_i^* \geq a_i + 1$. The branching progress stops when the assignment is not feasible or already in integer form or the optimal value is larger than previous upper-bound.

  This method is most widely used in *Mixed Integer Linear Programming* but not effective for our *0-1 Mixed Integer Linear Programming*. The reason is that we do not need step 1 to find $a_i.b_i$ since $a_i = 0$ for all variables here. ($b_i \in \{0, 1\}$).
- *Gomory Cuts*: This is an iterative method introducing new variables in each iteration. The main idea is that the new variables are some kind of residuals to characterize the gap between optimal assignment in linear programming and that in integer programming. This method may be useful to our problem. However, since our problem has $O(n^3)$ variables, the converge delay may be very long.

The indicator function f is not linear in last report. Here, we introduce n binary variables $Y = (y_1, ..., y_n)^T$ and for each y_t we have the constraint that

$$f(s^t) = y_t$$

$$\mathbf{1}^T s^t \mathbf{1} \leq n^2 y_t$$

$$y_t \in \{0, 1\}$$

So f is linear now.

I am still digesting these *Mixed Integer Linear Programming* methods now and try to get some substantial results in coming days.