Problem Solving in Computer Science
Course Notes : Lecture 18 (May 19^{th} 2005)
Scribe : Abhishek Garg
abishek.garg@epfl.ch

1 Groups Presentation

1.1 Group BGSa (*presented by Ali Salehi*)

They propose an algorithm which takes less number of rounds then the earlier proposed lower bound of $3b - n$. They do not provide the exact number of rounds which their algorithm will take. The algorithm works as follows. There is a $n^2 + n$ cells shared memory arranged in a matrix format of $(n + 1) \times n$. Row i and column i corresponds to cardinal i. In each cell, they store the ID of the cardinal. So, each cell is of the size $log\ n$. When the cardinal enters the shared memory for the first time, he sets the column corresponding to his ID to α and all the cells in the row to fis favourite. In the second access to the memory, the cardinal only sets the row to his favourite and do not touch the other cells in the column. The process continues until one of the cardinal sees all the elements in the column corresponding to his ID having non-α elements. At this time, this cardinal computes the winner and votes. He also sets the bit in the $(n + 1)^{th}$ row to 1. When the next cardinal comes in he sees this bit and computes the winner from this column and votes. This way, they claim that they can elect the leader in around $2b$ rounds.

There is a counter example raised by the BGS group, where the earliest any cardinal can vote is $2b - n$, so another b rounds are required for voting and hence $3b - n$ rounds in total. The counter example is same as the one their team faced in their approach for solving this problem with minimum number of rounds.

Finally everyone in the class agrees that the minimum number of rounds required for solving the Leader Selection and consensus problem is $3b - n$ in the worst case.

1.2 Group MST (*Presented by Gregory*)

They present an algorithm where they can take care of intelligent cheating with 2 bits of shared memory. In their earlier version of 2 bits shared memory algorithm, the leader computes the winner based on the votes of other cardinals and communicates the winner to other cardinals. This centralisation may lead to cheating on behalf of leader. To take care of this, they modify their algorithm so that the leader maintains the complete table of who voted for whom and transmits this complete table to each cardinal.

While transmitting this table, the leader doesn't know the ID of the cardinal to whom he is communicating the information, so receiving cardinal can verify his favourite in the table and can detect cheating on behalf of the leader.

2 Project 4 : Robotics

The 4^th project of this course is on designing a robot controller using the WEBOTS robot controller package. The problem is as follows. There are two robots in the maze, looking for a *red* object. Each robot is equipped with the following gadgets :

- **IR Sensors** Each robot has 6 Infrared sensors. They are used for computing the distance from the obstruction.
- **Camera** Camera is used for analyzing the colour of objects in front of robot. Camera has a angle of view of 60 degrees
- **Wheels** Every robot has 2 wheels to control the spped and direction of robot.
- **Battery** Each robot has a battery which is charged to maximum at the beginning of the game and dissipates with the movement of robot. The robot can recharge the battery from the chargers available in the world.

There are going to be two versions of the problem.

1. **Version 1 (Limited Battery)** Each robot has a limited battery, which is running out with the movements the robot makes. There are chargers in the maze, using which the robot can recharge his battery. So, the robot is looking for *red* object in the maze and at the same time he is also looking for chargers to recharge his battery. If, the robot runs out of battery then he is out of the game. If both the robot runs out of battery then the game is tied. Each robot can use any amount of memory for processing the data they collect from their sorroundings
2. **Version 2 (Limited Memory** In this version, in addition to the limit on battery, there is a limit on the internal memory of the robot. The goal is to design a robot with minimum amount of internal memory.

   The tutorial to build a simple robot using Webots can be found at www.cyberbotics.com.