Massachusetts Institute of Technology

16.410/413 Principles of Autonomy and Decision Making

Problem Set #10, Linear Programming using AMPL/CPLEX

Due in class, Monday, November 24, 2003.

Objective

To exercise your ability to formulate problems as linear programs over integer and real-valued decision variables. To encode and solve these problems using the AMPL/CPLEX system.

Problem 1 Astronaut Task Assignment

Write the mathematical program for each part of the following problem. Then write an AMPL program to solve it. Submit your AMPL program, and the output. Give the assignment of which astronauts will be assigned to what task?

Part 1. NASA has four tasks to be completed on the space station. There are four astronauts available to work on the tasks, but NASA's goal is to finish the entire list in the minimum time (because space walks are expensive). Because of radiation exposure concerns, each astronaut can complete only one task. Assume the tasks will be completed one at a time, that is, the objective is to minimize the total time spent.

	John	*Christa*	*Edward*	*Sally*
Adjust mirrors	37.7	32.9	33.8	37.0
Sample dust	43.4	33.1	42.2	34.7
Repair antenna	33.3	28.5	38.9	30.4
Polish fenders	29.2	26.4	29.6	28.5

Part 2. Now write an AMPL program assuming that all will spacewalk at the same time, so that the objective is to minimize the maximum of the task times. Again, submit your AMPL program, the AMPL run commands you used, and the list of astronauts assigned to each task.

Bonus points will be awarded if you can formulate both AMPL programs as linear programs, rather than integer programs. Give brief arguments: why is the linear program guaranteed to give a feasible assignment as the optimum? Why is solving as a linear program preferable to solving as an integer program?

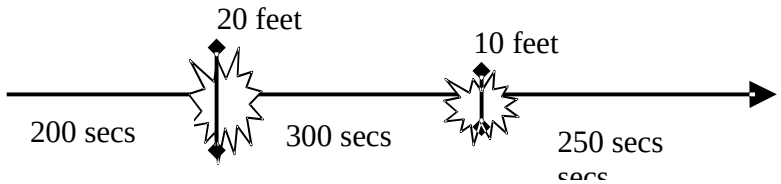
Problem 2 A Mars Rover obstacle avoidance problem

Part 1: Fixed Horizon obstacle avoidance

Design and implement in AMPL a trajectory planner for the Mars Rover, travelling near a line at x=0. The rover chooses a new setting for the heading angle φ relative to the line x=0 every 10 seconds. However, the new heading angle must be within 10 degrees of the previous heading angle or the Rover will roll. Use the small angle approximation to model this problem linearly. There are enormous dirt clods in the Rover's path, requiring that the car be displaced from the nominal path x=0 at 200 seconds by at least 10 feet, and at 500 seconds by at least 5 feet. The rover's speed is 0.5 feet/second. Use linear programming to design an autopilot for 700 seconds of roving, minimizing the absolute value of the deviation from the lane x=0 while avoiding the obstacles, assuming you know about all of the obstacles in advance.

Submit a description of your design, your AMPL program, and output demonstrating the correct function of your autopilot.

Note: The following AMPL constructs might be helpful in part 1:
set TIME := 1..70 ordered;
subject to Limit {j in TIME:j!=last(TIME)}:



Part 2. Receding Horizon Driving

In this section you will design and write pseudocode for a receding horizon Rover planner. One could write a looping function using AMPL commands (your model will be as above with a few changes) to handle the case where the planning is accomplished with a receding horizon of 200 seconds and a recalculation every 10 seconds. This might be necessary, for instance, if obstacles are not visible more than 100 feet ahead, or if the onboard computer is too slow to solve the 70 timestep version, or if observation of the obstacles and current heading is noisy so requires re-observation every 10 seconds.

However, since AMPL looping commands have arcane syntax and can not be run from the website, we are not asking you to put together a full AMPL implementation. Instead we are asking you to give a written description of how to modify your fixed horizon version in order to make it receding horizon. You should describe it in pseudocode, beginning with

```
for i in 1:70
```

and describing explicitly which constraints to change, which constants to change, when you will call `solve`, and which controls φ to actually apply to the rover.

Submit an explanation for your design changes, including the pseudocode, described above.

Part 3. Simplex algorithm

(from George Dantzig, Linear Programming and Extensions, Princeton University Press)

Solve using the simplex method the following linear program:

Maximize $Z=3y_1+4y_2$

Subject to: $2y_1 + y_2 \leq 2$

$y_1 - 2y_2 \leq 6$

$3y_1 + 9y_2 \leq 1$

$y_1 \geq 0,\ y_2 \geq 0$

Interpret each pivot step of the simplex algorithm geometrically in the plane of y_1 and y_2