

## Lab 3: Mixed Integer Linear Programming

Due: Wednesday 10/28/09 1:00pm

Professor A. Bayen

Fall 09

### 1 Lab overview

In this lab, you will learn how to use a very powerful industrial optimization software called CPLEX [2]. One of the features of CPLEX is to solve *Mixed Integer Linear Programs* (MILPs), i.e. linear programs in which some of the variables are continuous and some of the variables are discrete (decision variables). CPLEX is very tedious to use, so a modeling language called AMPL [1] has been designed to make engineers' lives easier. AMPL enables you to input MILPs into CPLEX in the exact same manner as you learned in class.

### 2 Using decision variables to solve non convex problems

In this first part of the lab, we are interested in a construction project in which  $N$  workers have to apply  $N$  layers of a protective coating to a new building on the Berkeley campus (each worker applies one layer). The  $N$  layers are required for the building to pass a certification test (i.e. be considered safe for the public). These  $N$  layers can be applied by the  $N$  workers in any order, but the effectiveness of the treatment depends on the time interval between the application of any two successive layers. In other words, in order to maximize the life expectancy of this treatment, it is best to wait as long as possible between the application of any two layers. You have to supervise the  $N$  workers and schedule their jobs. You can only afford to rent the machine needed to apply the coating for one day, so all  $N$  layers have to be applied within the same day. The workers are only working part time on this project, and each of them has different schedule constraints. The human

resources manager gives you the availability of each worker. Worker  $i$  is only available at the following times:

$$\text{Availability of worker } i = [a_{i,1}, b_{i,1}] \text{ or } [a_{i,2}, b_{i,2}] \text{ or } \cdots \text{ or } [a_{i,n_i}, b_{i,n_i}] \quad (1)$$

where  $n_i$  is the number of time intervals for which this worker is available.

**Question 2.1** We suppose that  $b_{i,j} < a_{i,j+1}$  for any  $i$  between 1 and  $N$ , and any  $j$  between 1 and  $n_i - 1$ . What does this mathematical condition represent in practice?

Your job is to find a time for each worker, called  $t_i$  for worker  $i$ , to achieve his task (we assume that the task is performed instantaneously). In order for  $t_i$  to be compatible with the worker's constraints, we need to make sure that  $t_i$  is in one of the intervals  $[a_{i,j}, b_{i,j}]$ .

**Question 2.2** Can the schedule constraints of the problem be implemented in the following form?

$$t_i \in [a_{i,1}, b_{i,1}] \cap [a_{i,2}, b_{i,2}] \cap \cdots \cap [a_{i,n_i}, b_{i,n_i}] \quad (2)$$

If this form is incorrect, explain why, and give the correct form of the schedule constraints.

**Question 2.3** We want to maximize the time between the application of any two layers of protective coating. Let us call  $T_{\text{paint}}$  this separation. The problem can be expressed as the following optimization problem.

$$\begin{array}{ll} \text{Maximize:} & T_{\text{paint}} \\ \text{Subject to:} & t_i \in [a_{i,1}, b_{i,1}] \cup [a_{i,2}, b_{i,2}] \cup \cdots \cup [a_{i,n_i}, b_{i,n_i}] \quad \forall i \in \{1, \dots, N\} \\ & |t_i - t_j| \geq T_{\text{paint}} \quad \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i \neq j \end{array} \quad (3)$$

Imagine that a computer program is able to solve problem (3). You have given a set of values  $N = 5$ ,  $a_{i,j}$  and  $b_{i,j}$  to the program, and the program returns  $t_1 = 10$ ,  $t_2 = 2$ ,  $t_3 = 4$ ,  $t_4 = 14$ ,  $t_5 = 8$ . The program does not return the value of the objective function  $T_{\text{paint}}$ . What is the value of  $T_{\text{paint}}$  in this example?

**Question 2.4** Can problem (3) be expressed easily in a linear manner as seen in class? If it cannot, enumerate all "difficulties".

**Question 2.5** Your office mate apparently found an equivalent way to express the same

problem:

$$\begin{array}{ll}
 \text{Maximize:} & T_{\text{paint}} \\
 \text{Subject to:} & t_i \geq a_{i,n_i} + \sum_{j=1}^{n_i-1} x_{i,j}(a_{i,j} - a_{i,j+1}) \quad \forall i \in \{1, \dots, N\} \\
 & t_i \leq b_{i,n_i} + \sum_{j=1}^{n_i-1} x_{i,j}(b_{i,j} - b_{i,j+1}) \quad \forall i \in \{1, \dots, N\} \\
 & x_{i,j} \leq x_{i,j+1} \quad \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, n_i - 2\} \\
 & x_{i,j} \in \{0, 1\} \quad \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, n_i - 1\} \\
 & t_i - t_j \geq T_{\text{paint}} - Cc_{i,j} \quad \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i > j \\
 & t_i - t_j \leq -T_{\text{paint}} + C(1 - c_{i,j}) \quad \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i > j \\
 & c_{i,j} \in \{0, 1\} \quad \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i > j
 \end{array} \tag{4}$$

You are not convinced that your office mate is right.

To understand the meaning of  $x_{i,j}$ , let us first consider an example for worker 2 (i.e.  $i = 2$ ) and  $n_2 = 8$ :  $x_{2,1} = 0$ ,  $x_{2,2} = 0$ ,  $x_{2,3} = 0$ ,  $x_{2,4} = 0$ ,  $x_{2,5} = 1$ ,  $x_{2,6} = 1$ . Given all the constraints in problem (4), what is the value of  $x_{2,7}$ ? Given all the constraints in problem (4) and the values of  $x_{2,j}$  provided earlier, in which interval does  $t_2$  lie?

**Question 2.6** Interpretation of  $x_{i,j}$  in the general case: in which situation do we have  $x_{i,j} = 0$ ? In which situation do we have  $x_{i,j} = 1$ ?

**Question 2.7** In the constraints involving  $T_{\text{paint}}$ , we assume that  $C$  is a large constant (as you have seen in class). Show graphically that the two sets of constraints

$$\begin{cases} t_i - t_j \geq T_{\text{paint}} - Cc_{i,j} & \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i > j \\ t_i - t_j \leq -T_{\text{paint}} + C(1 - c_{i,j}) & \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i > j \end{cases}$$

are equivalent to the set of constraints  $|t_i - t_j| \geq T_{\text{paint}}, \forall (i, j) \in \{1, \dots, N\}^2 \text{ s.t. } i \neq j$ .

### 3 Using CPLEX to solve the problem

Now that you are convinced that the optimization problem (4) effectively represents your problem, you can put it into AMPL/CPLEX and let CPLEX solve this for you.

A tutorial about AMPL and some example programs are available on the bSpace course website. These examples may be useful to show the AMPL syntax for parameters, vectors and matrices.

This section consists of four numerical examples for the optimization problem described in section 1 of the lab. Remember that all layers must be applied during the same day (between

5am and 10pm). We also assume that applying protective coating is instantaneous. For example, if a worker is available from 8am to 10am, we can choose 10am as the time at which this worker will apply protective coating, even though it is the very end of the period of availability of the worker.

**Question 3.8 – Very simple example.**

This first numerical example is very simple and is just meant to help you make sure your AMPL programs actually produces correct results. You DO NOT NEED to include this example in your report, nor submit the code for it.

There are two workers. Worker 1 is available from 8am to 11am, and worker 2 is available from 1pm to 3pm. At what time should each worker apply a layer of protective coating, so that the time between the application of each layer is maximum?

**Question 3.9 – A slightly more complicated example still solvable by hand.**

You can solve this example by hand, and make sure the results given by AMPL are correct. You DO NOT NEED to include this example in your report, nor submit the code for it.

There are three workers. Worker 1 is available from 9am to noon, worker 2 is available from 7am to 9pm, and worker 3 is available from 4pm to 6pm. At what time should each worker apply a layer of protective coating?

The following two examples should be included in the report, and the code should be submitted.

**Question 3.10 – Example 3.**

There are eight workers. The availabilities of the workers are the following:

Worker	Availabilities
1	8am-9am, 10am-10:30am, 2pm-2:30pm, 3pm-5pm
2	8am-9am, 9:30am-10:30am, 2pm-3pm, 4pm-5pm, 6pm-7pm, 8pm-9pm
3	7am-8am, 9am-12pm, 3pm-4pm, 5pm-7pm
4	8am-9am, 2pm-3pm, 4pm-5pm, 6pm-7pm
5	8:30am-9:30am, 2pm-2:30pm, 3pm-3:30pm, 5pm-5:30pm
6	8am-9:30am, 2pm-2:30pm, 3pm-4pm, 7pm-7:30pm
7	8am-10am, 10:30am-11am
8	2pm-2:30pm, 3pm-3:30pm, 4pm-5pm

At what time should each worker apply a layer of protective coating?

**Question 3.11 – Example 4.**

There are eight workers. The availabilities of the workers are the following (very close to example 3; the only difference is in **bold**):

Worker	Availabilities
1	8am-9am, 10am-10:30am, 2pm-2:30pm, 3pm-5pm
2	8am-9am, 9:30am-10:30am, 2pm-3pm, 4pm-5pm, 6pm-7pm, 8pm-9pm
3	7am-8am, 9am-12pm, 3pm-4pm, 5pm-7pm
4	8am-9am, 2pm-3pm, 4pm-5pm, 6pm-7pm
5	8:30am-9:30am, 2pm-2:30pm, 3pm-3:30pm, 5pm-5:30pm
6	8am-9:30am, 2pm-2:30pm, 3pm-4pm, 7pm-7:30pm
7	8am-10am, 10:30am-11am
8	<b>1pm-1:30pm</b> , 3pm-3:30pm, 4pm-5pm

At what time should each worker apply a layer of protective coating?

The data for example 3 and example 4 are very close. Indicate whether the results of the optimization for example 3 and example 4 are also close to each other, or not.

Which sentence do you think applies best to the optimization problem in these two examples:

1. The solution of the optimization is robust with respect to small changes in the input data.
2. The solution of the optimization is NOT robust with respect to small changes in the input data.

## References

- [1] R. FOURER, D. M. GAY, and B. W. KERNIGHAN. AMPL: a modeling language for mathematical programming. Boyd and Fraser, Danvers, MA, 1999.
- [2] ILOG CPLEX, <http://www.ilog.com/products/cplex/>.