CS358: Applied Algorithms

Assignment 7: ILP/MIP Project (due 12/05/25)

Instructor: Sam McCauley

Summary

This is a more open-ended assignment than the previous ones in the class. The goal of this project is to pick a difficult computational problem and solve it using (mixed) integer linear programming.

1 Example Problems

1.1 First Suggestion: Course Scheduling

If you don't want to come up with your own problem (or can't think of anything), I'd suggest the problem of scheduling classes for students at Williams.

In particular, each semester students select the classes they want to take. Then, the college decides what time(s) each class is and what room it is in. The goal will be to minimize the number of conflicts occurred by all students.

When constructing your instance, you may want to use the Class Blocks Grid for reference in terms of when courses may be scheduled: https://docs.google.com/spreadsheets/d/1TTGulGBTCtHH-8H9htY00D7drFN-50KkegKU3sPH6Mk/edit?usp=sharing. Note that different courses may meet for different numbers of times/amount of times during the week, and some of these times may overlap—you should take that into account in your schedule.

I don't know of a good reference of what classrooms are available at the college. I think it's OK if you make up some classrooms (but make sure they have different, and reasonable-looking sizes).

Be sure that your instance is nontrivial. For example, not all classrooms should be able to accommodate all classes, and some time slots should overlap.

1.2 Other Options

Here are some ideas from last year's iteration of the course to help you think of options:

- Scheduling NFL teams for a season. The goal is to schedule what games each team plays, and where the teams play them. Teams should have a good mix of home and away games, and each team should have a bye week. The goal is to minimize the total travel time incurred by all teams.

- Allocating camp counselors to camp activities. There are many activities at a children's camp, each occurring in potentially overlapping time slots. Some counselors need certifications to run certain activities, and some activities require the same resources and cannot be run concurrently. Counsellors also need a break between consecutive activities. The goal is to maximize the number of activities available over the whole schedule.

- Disaster relief allocation. After a disaster, many types of resources (e.g., food, water, medical supplies) need to be allocated to different affected regions, each within a certain specified time. These resources exist at a collection of supply centers; each supply center has a bounded amount of each resource. Each supply center is a certain distance (and, therefore, a certain amount of travel time) away from each affected area. How can we maximize the amount of resources delivered within the required times from the supply centers to the affected areas?

Bear in mind that your problem doesn't have to be real-world; almost any computational problem will work. For example, we could use MIP to solve the two towers problem that we saw on Assignment 1. (This does not turn out to be a good option here, as the underlying MIP is too simple (and it does not run well in practice). The idea is just that many—in fact, most by some definitions—computational problems can be phrased in this way.) One could similarly hope to encode, say, Sudoku puzzles in this way, or write an ILP that optimally loads a dishwasher for a given collection of dishes.

2 Problem Description

Problem 1. Please describe below the problem you are trying to solve. Be specific! What does a problem instance look like? What does a solution look like/what properties must it have? The description should be enough for a reader to construct an ILP/MIP themselves.

3 ILP Description

Problem 2. Below, please give an ILP or MIP to solve your problem. Your ILP/MIP should be a "recipe:" it should work for any instance of the problem.

Solution. □

4 Code file

Problem 3. Below, please give the specific instance of your problem that you solved in the `.lp` file. Your instance should be nontrivial—it should have a reasonably large number of constraints (let's say: comfortably more than 5–10), and you should try to avoid trivial solutions (like in the example below: if our ILP schedules courses in classrooms, not every course should be scheduleable in every classroom).

Solution. □

Implementation 1. Please submit an `.lp` file in your git repo that solves the example instance you gave above.