Technische Universität Berlin Prof. Andreas Bley
WiSe 2012

Computational mixed-integer programming

5th set of programming exercises:
Lagrange Relaxation for the OPMPSP

This exercise shows how to implement a basic Lagrange Relaxation. We will use our own subgradiant method to find good estimates for the Lagrangian Dual variables and thus a good lower bound to the OPMPSP. We will then compare the bounds for different relaxations.

Problem Description

We consider the ILP formulation for OPMPSP that has been introduced in exercise 3:

$$\max \sum_{t=0}^{T-1} \delta^t \sum_{i=0}^{N-1} (p_i - m_i)(x_{t,i} - x_{t-1,i})$$

$$\text{s.t.} \quad x_{t-1,i} \leq x_{t,i} \qquad \forall\, i,t \tag{1}$$

$$\sum_{i=0}^{N-1} R_i x_{t,i} \leq (t+1)M \qquad \forall\, t \tag{2}$$

$$\sum_{i=0}^{N-1} O_i x_{t,i} \leq (t+1)P \qquad \forall\, t \tag{3}$$

$$x_{t,i} \leq x_{t,k} \qquad \forall\, t,i,k \in \mathcal{P}(i)$$

$$x_{t,i} \in \{0,1\} \qquad \forall\, t,i$$

Check the third exercise sheet, if you forgot what the meaning of the variables and the exact denotation was. As in the third exercise, we assume that this mathematical model also contains variables $x_{i,-1} = 0$ for all i, which are, of course, not generated in the implementation of the model.

Getting started

The given package `exercise05.tgz` contains all necessary files for this exercise. Amongst others, it contains:

`src/main.cpp` This is the main source file. Here, nothing needs to be implemented by you.

`src/lagrange_capacity.cpp` contains the implementation of the Lagrangian relaxation where the capacity constraints (2) and (3) are relaxed. Your first task is to complete the implementation in this file.

`src/lagrange_time.cpp` contains the implementation of the Lagrangian relaxation where the time-coupling constraints (1) are relaxed. You second task is to complete the implementation in this file.

`data/*` Test instances to test your implementation.

As usual, you can compile your files using `make` after creating the softlink to the local scip installation in subdirectory `lib`. The binary can be executed as
`bin/opmpsp {1,2,3} data/rand035.blocks data/rand035.arcs`,
where the first parameter controls if the standard ILP approach, the capacity relaxation or the time-linking relaxation will be run.

1. Relaxation of the Processing and Mining Constraints

In this subtask, we implement the Lagrangian algorithm where the constraints (2) and (3) are relaxed. Take a look at the file `lagrange_capacity.cpp`.

(a) First, a SCIP model containing only the non-relaxed constraints is created. Note that the objective function of this created model is exactly the objective function of the original problem. Any additional terms you need here for the Lagrangian relaxation have to be added or changed by you.

(b) Find the first `BEGIN_TODO`. Here you need to create the dual variables for the relaxed constraints (2) and (3) and initialize them (to 0, for example).

(c) Implement the main `for`-loop. In this loop you have to

- solve the relaxed problem
- extract its solution
- compute the Lagrangian objective value
- compute the violation of the relaxed constraints (2) and (3)
- compute the norm of the violation vector, and finally
- update the dual variables and the objective function the relaxed problem.

Parts of these steps and the necessary data structures are already given in the file. Fill out the gaps between the `BEGIN_TODO` and `END_TODO`'s.

Note that the objective of the relaxed problem is

$$\max \sum_{t=0}^{T-1} \delta^t \sum_{i=0}^{N-1} (p_i - m_i)(x_{t,i} - x_{t-1,i}) - \sum_{t=0}^{T-1} \pi^t \left(\sum_{i=0}^{N-1} R_i x_{t,i} - (t+1)M \right) - \sum_{t=0}^{T-1} \mu^t \left(\sum_{i=0}^{N-1} O_i x_{t,i} - (t+1)P \right)$$

where $\pi^t \geq 0$ and $\mu^t \geq 0$ are the Lagrangian duals for the mining capacity and processing capacity constraint of period t, respectively. Thus, the objective coeffiecient of the x-variables in the relaxed (remaining) problem need to be adjusted when the duals vary and, in order to compute the overall Lagrangian objective, also the cost term

$$+ \sum_{t=0}^{T-1} \pi^t (t+1)M + \sum_{t=0}^{T-1} \mu^t (t+1)P$$

induced by the right hand sides of the relaxed constraints needs to be considered. (Steps 6 and 2 in the main for loop).

Use the *harmonic series* to adapt the step size in each iteration, i.e. `stepsize` $:= \frac{1}{\texttt{iteration}+1}$. Do not forget to output the solution value of the current iteration to the console so that you can actually see how the bound evolves over time .

(d) Test your model on the small `rand035` instance. You should get a best bound close to (but not better than) the value of the LP relaxation of the original problem. You get the LP relaxation value by running the executable program with first parameter 1; the LP relaxation value will the be printed out when the root relaxation of the ILP model is solved.

2. Relaxation of the Time Linking Constraints

Now implement a Lagrangian algorithm where you relax the constraints (1), while keeping all other constraints.

Compare its result with the bound you got from the first relaxation.

The implementation of the time-linking relaxation works analogous to that of the capacities relaxation. One of the advantages of the time-linking relaxation, however, is that the only constraints linking different time periods are now removed. That means that the relaxed problem decomposes into T many small IP problems. Instead of setting up one single large IP you should set up T independent IPs, each corresponding to only one time period t and solve them independently (either sequetially one after the other or in parallel).

Again, parts of the implementation and the necessary data structures are already given in the file. You only need to fill out the gaps between the `BEGIN_TODO` and `END_TODO`'s.

After completing the implementation, answer the following questions:

(a) Why might the processing of T smaller IPs be more efficient than solving the whole IP?

(b) Why is the bound obtained by the time-linking relaxation tighter (lower) than the bound obtained by the capacity relaxation?