Report Multi-agent Systems Module III

Tom Jacobs (s0214835), Thomas Uyttendaele (s0215028)

May 31, 2013

Contents

1 First Assignment

1.1 Earliest & latest starting time

The first question focusses on using the `FastFloyd` algorithm to determine the earliest and latest starting times of the events specified. An implementation can be found in the `ex1.m` source file together with the `LoadMatrix` function. The minimal timings are the negated values of the first column of the STN after using the `FastFloyd` algorithm, while the first row represents the maximum timings.

1.2 Constraints

To add arbitrary STN-constraints the `newSTN` function is defined. It receives a constraint in the parameters and verifies that the constraint can be added without creating an inconsistency. This is done by checking that the new constraint isn't tighter than the minimal value already there.
e.g. The contraint $t_i - t_j \leq \delta$ is allowed if $\delta \geq -STN_{i,j}$.
If this is true and δ is smaller than the current value of $STN_{j,i}$, the constraint is set in the matrix and the `FastFloyd` algorithm is applied again. To allow more efficient

processing later on a matrix of constraints can also be entered that are checked in their entirety. In that case, newly introduced inconsistencies are detected by checking if the diagonal elements of the STN remain zero after running `FastFloyd`. An example testing the implementation can be found as `ex2.m`, that shows inconsistent and consistent constraints added to `stn5` and `snt49`.

1.3 Iterative schedule construction

To allow the construction of arbitrary schedules, an STN is loaded and the user is prompted for input. Should the allowed range of inputs have a minimum and maximum value that are the same, the user is informed that this value is chosen by default and thus doesn't need to enter it himself. This is done in `ex3.m`. If the value entered isn't inside the allowed interval it is refused, and otherwise added to the resulting matrix and the `FastFloyd` algorithm is run to ensure the STN is updated. Afterwards the column and row of the chosen event are removed from the matrix. This is allowed as running the `FastFloyd` algorithm ensures the newly applied changes have been propagated across the STN. Appendix A.1 shows a trace for the problem `stn49.tab`.

1.4 Schedule monitoring

For this question, a schedule is being checked as input step by step chronologically against the STN provided. Every iteration it is verified that the input value lies between the minimal and maximal values specified by the STN. Should this value be allowed, it is entered into the STN using the method from section 1.3. As soon as a constraint is violated, the process stops and determines that the current input is invalid. Otherwise, the schedule is accepted as valid. `ex4.m` provides an implementation with a correct and incorrect input of size 145 exported in table form, that are equal except for a value of 0 on a_{42}. Switching between these two can be done by choosing either *correct145.tab* or *incorrect145.tab* on line 5 of the script.

2 Second Assignment

2.1 Scheduling airplanes

2.1.1 Determining the number of runways

To calculate the number of runways required, the function `calculateNbOfRunways-WithThreshold` is created with an STN modelling the problem. The function expects a `threshold` parameter, that can be interpreted as the amount of runways that are overall available. It starts by creating an STN for the problem and running the `FastFloyd` algorithm. Then finds an assignment of the planes over the runways using earliest possible scheduling on their take-off time.

The first peak (a peak being a timeslot where more planes are departing than runways available as the threshold) chronologically encountered is to be eliminated. All planes from this peak are sorted by their last possible take-off time, because their scheduling is the most flexible. Should two planes have an equal latest take-off time their ordering by earliest take-off time is used. The first *threshold* number of planes is kept, and all others in the peak get an additional constraint of coming at least 5 minutes after the first plane in that peak. These constraints are added using the method specified in section 1.2.

Number of Runways	3	4	207	-
Flexibility	4687	6112	11371	11371

Table 1: The maximum flexibility for different amounts of runways. *The final column represents the theoretical maximum flexibility achievable.*

This is repeated either until the STN becomes invalid or no peaks remain. If it is invalid no solutions are possible with the chosen threshold, otherwise, the calculated STN is returned and a possible solution can be found by negating its first column.

With the example supplied, the algorithm fails with thresholds 1 and 2, respectively at time points **10:55** and **15:23**. At first success with threshold 3 the solution found is exported for later use. To achieve this solution 11 new sets of constraints had to be added. For practicality, this is repeated with thresholds 4 and 207, where it of course succeeds as well, without the need for any added constraints. This implies that even with 207 runways only 4 will be used at the same time. It will later be shown however, that while at first sight these thus appear similar, this still has serious implications for the flexibility of the schedules. `ex5.m` provides an implementation for calculating the minimal amount of runways.

2.1.2 Flexibility with minimal runways

The first assignment found during peak shaving with the STN constructed in section 2.1.1 in an attempt to use only 3 runways is the one used from here on. This is thus by definition the earliest possible assignment on 3 runways.

Calculation of the flexibility can be done in two ways. Either based on the STN itself, or the one preferred here using the earliest departure and take-off times from the original input and earliest take-off times from the calculated STN. From the solution found earlier, only the order in which the planes leave is kept. For each individual plane the following constraints are created.

- $-t^-_{dep} \leq -earliestDepTime$
- $t^+_{takeoff} \leq latestTakeoffTime$
- $t^+_{dep} - t^-_{takeoff} \leq -20$
- $t^-_{dep} - t^+_{dep} \leq 0$
- $t^-_{takeoff} - t^+_{takeoff} \leq 0$

Relations between planes are then described as such, where i is the current plane and $i + threshold$ is the number of the plane for which the constraint is to be added:

$$\forall i \in 1..(\#_{planes} - threshold) : t^+_{takeoff,i} + 5 \leq t^-_{takeoff,i+threshold} \quad (1)$$

This enforces that each further plane is is placed at least 5 minutes later, resolving the peak. Using the `linprog` function in MATLAB the minimum of a defined target function, being the maximum flexibility, can be found. This is not necessarily an optimal solution.

This question itself is solved by the first part of `ex5b.m`. The flexibility that this schedule has can be found in table 1.

Constraint	New Value	Old Value
$TP^C_{ET} - z_0 \leq$	600	720
$z_0 - TR^A_{ST} \leq$	-600	-480
$TR^A_{ST} - z_0 \leq$	603	720
$z_0 - TP^C_{ET} \leq$	-594	-480
$R^A_{ST} - z_0 \leq$	480	720
$z_0 - R^B_{ST} \leq$	-480	-480
$R^B_{ST} - z_0 \leq$	480	720
$z_0 - R^A_{ST} \leq$	-480	-480

Table 2: Overview of the new constraints introduced in section 2.2.3

2.1.3 Flexibility comparison scenario's

Similarly to the previous question the rest of `ex5b.m` executes the algorithm again for 4 and 207 runways. The input for 4 runways is provided again by the result of section 2.1.1, giving us the flexibility listed in table 1. This 30.4% flexibility improvement from 3 to 4 runways was expected. It is also clearly visible that the flexibility will grow asymptotically to its maximum value reached at 207 runways.

Just to prove the correctness of the algorithm, the flexibility of a schedule with 207 runways is calculated as well. This turns out to be the exact same value as one would get from simply summing all the intervals in the original `standplan.txt`, and subtracting 20 minutes everywhere to account for the difference between departure and take-off time.

2.2 Planning for 3 people

2.2.1 Flexibility of the STN

The STN is constructed from the paper provided and a specific function `determineFlexFromSTN` has been implemented to determine the flexibility given a certain STN. For every value ($STN_{i,j}$) not equal to 10000 or on the diagonal, the constraint $t_j - t_i \leq STN_{i,j}$ is added. After solving this problem with `linprog`, the flexibility is determined to be 180, as implemented in `ex6a.m`

2.2.2 Flexibility of the decoupled version

Analogously to section 2.2.1, the flexibility found by `ex6b.m` is now 135. This drop of 45 minutes is due entirely to a reduction in *Bill*'s flexibility. The reduction process forced activity R^B_{ST} to the point in time right in the middle of its previous interval. From here on the flexibility later in time is thus reduced from 120 tot 75 minutes.

2.2.3 Flexibility preserving decoupling

To achieve a better decoupling the flexibility is again determined as the solution for maximum flexibility. There exist 4 different constraints between the partitions of the problem. Each of these constraints ($t_i - t_j \leq STN_{i,j}$) can be split into two subconstraints $t_i - z_0 \leq v^+_i$ and $z_0 - t_j \leq -v^-_j$. Here the v values are the lower and upper bounds of the flexibility interval. The results of this entire process (which is determined in `ex6c.m`) are the constraints in table 2

2.2.4 Decoupling for even distribution of flexibility

The regular algorithm to solve this problem is described in algorithm 1. Our code can be found in `ex6d.m`. A manual implementation of an easier algorithm was done as the problem at hand provides some shortcuts. The maximum flexibility for each party is calculated with the added constraint that the original maximum has to be kept, and it is determined that the flexibility for *Chris* and *Ann* is limited to 30 and *Bill*'s is found as 120. By the structure of the implementation, no extra iterations are needed. Also, no better distribution is possible. Using the `linprog` function to maximise the global flexibility with as extra constraint the here described local flexibilities.

The extra constraints can again be derived as explained in section 2.2.3, for another 8 extra constraints. Interparty connections can now be removed. `ex6d.m` provides an implementation for this solution.

```
Data: TotalFlexibility: Flexibility for the whole problem
Data: PartitionSet : Set of all partitions
Data: FinalPartitions : The set of all partitions, initially
      empty
Result: ExtraConstraints : The set of all extra constraints
        based on individual flexibility

Calculate TotalFlexibility;

while PartitionSet not empty do
    ThresholdFlexibility = $\frac{TotalFlexibility}{\#PartitionSet}$ ;
    forall the Partition in PartitionSet do
        if $Flexibility_{partition} \leq ThresholdFlexibility$ then
            Remove Partition from PartitionSet;
            Add Partition to FinalPartitions ;
            Add Constraint $TargetFunction_{Partition}$ =
            $Flexibility_{Partition}$ to ExtraConstraints;
            TotalFlexibility = TotalFlexibility -
            $Flexibility_{Partition}$;
        end
    end
end
```

Algorithm 1: Algorithm for decoupling while preserving maximum flexibility

A Examples and Traces

A.1 Trace for matrix size 49 (Question 1.3

```
Choose a value for variable 41 in the range of -627 and 791?       658
Choose a value for variable 7 in the range of -599 and 1065?       921
Choose a value for variable 32 in the range of 392 and 1787?       528
Choose a value for variable 15 in the range of -782 and 1049?      219
Choose a value for variable 48 in the range of 240 and 1380?      1340
Choose a value for variable 9 in the range of -1422 and -315?     -347
Choose a value for variable 47 in the range of -1355 and -255?    -821
Choose a value for variable 38 in the range of -1285 and -1149?  -1266
Choose a value for variable 21 in the range of -209 and 127?        99
Choose a value for variable 37 in the range of -366 and 681?       639
Choose a value for variable 30 in the range of -979 and 459?      -928
Choose a value for variable 42 in the range of -1036 and -224?    -277
Choose a value for variable 31 in the range of 594 and 1109?       984
Choose a value for variable 35 in the range of 356 and 1113?       653
Choose a value for variable 28 in the range of 602 and 1602?       773
Choose a value for variable 33 in the range of -475 and -422?     -474
Choose a value for variable 12 in the range of -690 and -650?     -689
Choose a value for variable 5 in the range of 433 and 1149?       1023
Choose a value for variable 29 in the range of 235 and 263?        244
Choose a value for variable 46 in the range of 101 and 227?        105
Choose a value for variable 19 in the range of -259 and 364?       -21
Choose a value for variable 36 in the range of -188 and 634?       466
Choose a value for variable 8 in the range of 676 and 798?         736
Choose a value for variable 20 in the range of -1330 and -579?    -844
Choose a value for variable 34 in the range of 724 and 1034?       958
Choose a value for variable 13 in the range of -992 and -69?      -364
Choose a value for variable 26 in the range of 322 and 323?        322
Choose a value for variable 4 in the range of 300 and 332?         316
Choose a value for variable 49 in the range of -955 and 733?      -381
Choose a value for variable 24 in the range of 290 and 466?        329
Choose a value for variable 39 in the range of 1009 and 1086?     1028
Choose a value for variable 18 in the range of -814 and -670?     -713
Choose a value for variable 44 in the range of 597 and 1020?      1003
Choose a value for variable 22 in the range of -753 and -388?     -703
Choose a value for variable 6 in the range of 439 and 1223?        641
Choose a value for variable 40 in the range of -290 and -232?     -275
Choose a value for variable 27 in the range of -795 and -485?     -720
Choose a value for variable 45 in the range of -891 and -874?     -885
Choose a value for variable 3 in the range of 141 and 201?         156
Choose a value for variable 17 in the range of -962 and -923?     -944
Choose a value for variable 11 in the range of 147 and 153?        152
Choose a value for variable 23 in the range of -1059 and -408?    -701
Choose a value for variable 43 in the range of -339 and -338?     -339
Choose a value for variable 16 in the range of -895 and -232?     -395
Choose a value for variable 10 in the range of -1644 and -883?   -1212
Choose a value for variable 2 in the range of 224 and 1223?        277
Choose a value for variable 25 in the range of -400 and -212?     -253
Choose a value for variable 14 in the range of -1600 and -964?   -1518
```

The chosen values are: 0 277 156 316 1023 641 921 736 -347 -1212 152 -689 -364 -1518 219 -395 -944 -713 -21 -844 99 -703 -701 329 -253 322 -720 773 244 -928 984 528 -474 958 653 466 639 -1266 1028 -275 658 -277 -339 1003 -885 105 -821 1340 -381