

DM204 - Scheduling, Timetabling and Routing

Exam Project, Fall 2009

Problem B

1 Practical Notes

Note 1 The project consists in modeling a real-life problem, designing a solution algorithm, implementing a solver, experimenting with some real or synthetic instances of the problem and documenting the work in a written report.

The evaluation of the project is based on the written report only, which may be in English or in Danish. However, a program that implements the solver described in the report must also be delivered. The program can be used to check the correctness of the solutions discussed.

The grade of the project will contribute to 40% of the final grade of the course. There is no external censorship on the project.

Note 2 Additional material to the project description, like input data, is available on the web at:

<http://www.imada.sdu.dk/~marco/DM204/>. Corrections or updates to this project description or to the web page above will be posted in the announcement list of the Blackboard system.

Note 3 *Submission.* An archive containing the electronic version of the written report and the source code of the program must be handed in through the Blackboard system **before 14:00 of Monday, June 19, 2009**. This is the procedure:

- go at the course page in the Blackboard system;
- choose "Exam Project Submission" in the menu on the left;
- fill the form and conclude with submit;
- print and keep the receipt (there will be a receipt also per email).

See Appendix C for details on how to organize the electronic archive. Reports and codes handed in after the deadline will generally not be accepted. System failures, illness, etc. will not automatically give extra time.

2 Problem Description

Problem Statement The trucks of the transportation company "Move Efficiently" have to undergo periodic maintenance of different types ranging from changing the oil to a complete engine overhaul. The time intervals between consecutive maintenance checks

of each type are pre-described based on experience and information from the truck manufacturer. The trucks become unusable if they are behind on the required maintenance. However, maintenance is expensive and trucks should not be over-maintained. The problem, therefore, is to schedule maintenance so that it minimizes costs while making sure that trucks can be used. The problem needs to be solved over a 2-year time-horizon on a weekly basis, From July 1, 2009 till June 30, 2011. There are constraints on the number of trucks that are required to be active, and capacity constraints at the single maintenance location.

Parameter Values

- Number of trucks: 425
- Number of truck types: 3 (Small, Medium, Large)
- Different Types of Maintenance:
 - Routine Maintenance
 - Transmission Maintenance
 - Engine overhaul
- Maximum interval between "Routine Maintenance": 15 weeks
- Maximum interval between "Transmission Maintenance": 50 weeks
- Maximum interval between "Engine overhaul": 90 weeks

During an "Engine overhaul" the "Routine Maintenance" and "Transmission Maintenance" are also performed. During "Transmission Maintenance" the "Routine Maintenance" is also performed.

- The cost of a "Routine Maintenance" for any truck type at any time: \$250
- The cost of a "Transmission Maintenance" for any truck type at any time: \$2,000
- The cost of an "Engine overhaul" for any truck type at any time: \$7,500
- Duration of "Routine Maintenance": 0.5 week
- Duration of "Transmission Maintenance": 1 week
- Duration of "Engine overhaul": 3 weeks

The vehicle is out of order for the duration of the service.

At the beginning of the time period the maintenance level of the trucks is roughly uniformly distributed. See the exact data in the instance available. As expected, this part of the data is very likely to change. The total number of trucks, the minimum number of trucks in service and the maintenance capacities are reported in Table 1.

Solution The solution is a maintenance schedule, that is, a schedule of maintenances for each truck over the time horizon.

	All trucks	Small	Medium	Large
Total number of trucks	425	85	295	45
In-service minimum number of trucks	400	75	280	40
Capacity of "Routine Maintenance"	11	3	8	2
Capacity of "Transmission Maintenance"	4	1	3	1
Capacity of "Engine overhaul"	15	3	11	2

Table 1: Trucks available, operating demand and capacity constraints of the maintenance facility.

Your task Your task is the following:

- Model mathematically the problem defining the variables, the constraints and the objectives. If this is possible, find a mixed integer programming (MIP) formulation.
- If a MIP model is produced, write a ZIMPL program for it and solve the instances available using a time limit of 5 hours.
- If the MIP formulation turns out to be infeasible either in modelling terms or in terms of computation times, try using other approaches of your choice (e.g., constraint programming, heuristics) to achieve at least a feasible solution.
- Analyze and discuss experimental results on the instances available.

It is not necessary to accomplish all the points above to achieve the highest grade in the project. However, if the problem turns out to be computationally easy to solve, then you can consider the following questions that the company is pondering to decide whether to expand the maintenance facility:

- Expenses aside, how many trucks can be kept on the road continuously, given the current capacity of the maintenance facility?
- Where are the bottlenecks in the current system? How should the maintenance station be extended?

Suggestions and hints Start with a smaller number of trucks to see if your formulation works. Try a shorter time horizon or a longer time unit. The complete model may take up to 10 hours to solve. Avoid hard-wiring numbers in your model. If the problem turns out to be infeasible for a certain dataset (i.e., the required number of trucks can not be guaranteed for some weeks), make sure your model still returns something that can be implemented, such as maximize the number of available trucks, while keeping the cost minimal.

3 Remarks

Remark 1 The following criteria will be kept in consideration in the evaluation:

- Good model structure, keeping the model as simple as possible.
- The quality of the solution found and the solution time.
- Clear reporting with focus on the relevant key performance.

Remark 2 If you find issues that are not specified in full detail make your own assumptions, i.e., try to guess what is probably happening in practice, describe the way how you suggest the issue should be handled, and formulate the models according to your version of the anodizing plant problem.¹

Few issues have been purposely left out from the description but are present in practice. Tanks are grouped according to type of treatment and a stage in the recipe can be accomplished in any of the tanks that offer the treatment required. Maximizing the throughput is not the only criterion to evaluate a schedule. There might be priorities assigned to the bars or due dates and deadlines. Finally, it might be necessary to avoid that some bars remain too long in the buffer line. These issues should not be considered in the project.

Remark 3 In case the solution is found by exact methods, the source code of the model must be delivered and it must be stated which program is used to solve it (e.g., CPLEX, SCIP, GECODE, etc.). Only freeware software is allowed. For heuristic algorithms, the programs must output the solution in a reasonable amount of time. Solutions must be written in a file in text format as described in Appendix B.

Remark 4 The total length of the report should not be less than 10 pages and not be more than 15 pages, appendix included (lengths apply to font size of 11pt and 3cm margins). Although these bounds are not strict, their violation is highly discouraged. Do not include source code in the report.

Acknowledgments This document is a copy of the description at the AIMMS-CPLEX/MOFTA Optimization Modeling Competition 2009, published by the Department of Industrial & Systems Engineering, Lehigh University. 2009.

¹This situation is very typical in practice. In the meetings with the company, the engineers might have forgotten to tell some details that then become necessary while modelling the problem. Often for reasons of time, these issues must be resolved by making some additional reasonable assumptions. Hopefully, then practitioners agree. Otherwise something must be changed.

Appendix A Instance Format

At the beginning of the project there is only one instance available. More instances, up to 5 in total will become available during the project. The data are organized in a csv file. The first column reports an identifier number for the Truck taken in $\{1, 2, \dots, n\}$ where n is the total number of trucks. The second column indicates the Category of the truck. The other three columns report the “Weeks since last engine”, the “Weeks since last transmission” and “Weeks since last routine”.

Appendix B Solution Format

A solution should report all the maintenances scheduled and should be written in a csv or a text file. In the file there is a line for each maintenance with columns that represent:

- the truck identifier
- the type of maintenance (use the encoding 1 = Routine, 2 = Transmission, 3 = Engine).
- the week number in which the maintenance starts.

Appendix C Electronic Submission

The electronic submission must be organized in an archive that expands in a main directory named with your full CPR number and with content

```
<CPR-NUM>/README  
<CPR-NUM>/report/  
<CPR-NUM>/src/  
<CPR-NUM>/zimpl/
```

The file README contains the instructions for compiling and running the solver. The directory src contains the sources which may be in C, C++, JAVA or other languages. If needed a Makefile can be included either in the root directory or in src. After compilation the executable must be placed in src. For JAVA programs, a jar package can also be submitted. The directory zimpl contains models to be interpreted by ZIMPL.

Programs must work on IMADA’s computers under Linux environment and with the compilers and other applications present on IMADA’s computers. Students are free to develop their program at home, but it is their own responsibility to transfer the program to IMADA’s system and make the necessary adjustments such that it works at IMADA.²

²Past issue: make sure the JAVA compiler and virtual machine are the ones you intend to use (check the path: `ls -l /etc/alternatives/javacpath`); in C, any routine that uses subroutines from the math.c library should be compiled with the `-lm` flag – eg, `cc floor.c -lm`.