Chalmers University of Technology
University of Gothenburg
Mathematical Sciences
Optimization
Ann-Brith Strömberg
Caroline Granfeldt

MVE165
MMG631
Linear and integer optimization
with applications
Assignment information
May 6, 2019

Assignment 2: Maintenance scheduling

Implementations of the model in Julia are found on the course homepage:

`www.math.chalmers.se/Math/Grundutb/CTH/mve165/1819/`

Study the Julia files carefully to get some hints before you start solving the exercises. The file `as_run.jl` should be edited in order to solve the different instances of the model, as described in the exercises below.

The questions 1–3 below are mandatory. In addition, students aiming at grade 3 or G must answer *at least one* of the questions 4–5, while students aiming at grade 4, 5, or VG must answer *all* the questions.

The mathematical model

Given is a mathematical model for finding a maintenance schedule such that the costs of maintaining a system during a limited time period is at minimum, while ensuring that the system is functioning during the entire period. The system consists of several components with economic dependencies and limited lives. The mathematical model is developed in the article[1] by Almgren et al. (2012), which also describes the problem background; the model is also described in the notes of Lecture 8b.

Sets and parameters

- $\mathcal{N}$ = the set of components in the system. `(in Julia: Components)`
- T = the number of time steps in the planning period. `(in Julia: T)`
- T_i = the life of a new component of type $i \in \mathcal{N}$ (measured in number of time steps). It is assumed that $2 \leq T_i \leq T - 1$. `(in Julia: U)`
- c_{it} = the cost of a spare component of type $i \in \mathcal{N}$ at time t (measured in €). For some instances it is assumed that c_{it} is constant over time, i.e., $c_{it} = c_i$, $t = 1, \ldots, T$. `(in Julia: c)`
- d_t = the cost for a maintenance occasion at time t (measured in €). For some instances it is assumed that d_t is constant over time, i.e., $d_t = d$, $t = 1, \ldots, T$. `(in Julia: d)`

[1] T. Almgren, N. Andréasson, M. Patriksson, A.-B. Strömberg, A. Wojciechowski, and M. Önnheim: *The opportunistic replacement problem: theoretical analyses and numerical tests*, Mathematical Methods of Operations Research, vol. 76, no. 3, pp. 289–319, 2012. Reachable from Chalmers' domain, at `http://dx.doi.org/10.1007/s00186-012-0400-y`

Decision variables

- $x_{it} = \begin{cases} 1 & \text{if component } i \text{ is replaced at time } t, \\ 0 & \text{otherwise,} \end{cases} \qquad i \in \mathcal{N},\ t \in \{1, \dots, T\}.$
- $z_t = \begin{cases} 1 & \text{if maintenance is made at time } t, \\ 0 & \text{otherwise,} \end{cases} \qquad t \in \{1, \dots, T\}.$

The model

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} \left(\sum_{i \in \mathcal{N}} c_{it} x_{it} + d_t z_t \right), && && \text{(1a)} \\
\text{subject to} \quad & \sum_{t=\ell+1}^{\ell+T_i} x_{it} \geq 1, && \ell = 0, \dots, T - T_i,\ \ i \in \mathcal{N}, && \text{(1b)} \\
& x_{it} \leq z_t, && t = 1, \dots, T,\ \ i \in \mathcal{N}, && \text{(1c)} \\
& x_{it}, z_t \in \{0, 1\}, && t = 1, \dots, T,\ \ i \in \mathcal{N}. && \text{(1d)}
\end{aligned}
$$

Description of the model

(1a) The objective is to minimize the total cost for the maintenance during the planning period (the time steps $1, \dots, T-1$) `(in Julia: Cost)`.

(1b) Each component i must be replaced at least once within each T_i time steps `(in Julia: ReplaceWithinLife)`.

(1c) Components can only be replaced at maintenance occasions `(in Julia: ReplaceOnlyAtMaintenance)`.

(1d) All the variables are required to be binary.

Exercises to perform and questions to answer

1. (a) Solve the model (1) as implemented in the file `as_mod.jl` with data from `as_dat_large.jl`, letting $T = 125$, and with integer requirements on the variables x_{it} and z_t. *Note that in this instance all costs are time independent, i.e.,* $c_{it} = c_i$ *and* $d_t = d$, $t = 1, \dots, T$. Relax the integrality requirements on the variables x_{it} and resolve the problem. Then relax the integrality on all variables and resolve the model. Compare the solutions obtained and discuss their interpretations. Compare also the computation times (CPU) and explain the differences.

   (b) Solve the model (1) as implemented in `as_mod.jl` with data from `as_dat_small.jl`. Relax the integrality constraints on the variables and resolve. Then add the constraint given by the last function in the file `as_mod.jl` by calling (from the run-file) the function `add_cut_to_small(m)` and resolve. Compare the solutions obtained and explain their differences.

   (c) Prove that the additional constraint implemented in `as_mod.jl` is a *valid inequality* (i.e., it does not cut away any feasible solution) to the instance of (1) defined by `as_dat_small.jl`.

2. Solve the model (1) as implemented in `as_mod.jl` and `as_dat_large.jl`, letting $d = 20$, and with integer requirements on the variables z_t only.

   (a) Vary the time horizon between $T = 50$ and approximately $T = 200$ and draw a graph of the computing time (in CPU seconds) as a function of T (use a log-scale). If needed, use the options for limiting the size of the branch–and–bound tree—keeping track of upper and lower bounds on the optimal value (see the file `as_run.jl`).

   (b) Make an analogous graph for the case when the integrality requirements on the variables are relaxed; vary the time horizon between $T = 50$ and approximately $T = 700$.

   (c) Compare and comment on the complexity properties of the two models solved in 2a and 2b.

   (d) Gurobi uses the branch–and–bound algorithm, possibly employing presolve steps including heuristics and cutting plane generation. On what does it seem to spend most of the solution time: presolve, finding an optimal (feasible) solution, or verifying its optimality?

3. Implement in Julia the model (1a)–(1f) from the article[2] by Gustavsson et al. (2014).

   (a) Repeat the tests made in exercise 2a–2c for this model. Illustrate the results with suitable graphs.

   (b) Compare the outcomes from the two models and make relevant conclusions. Illustrate with suitable graphs.

   (c) Discuss the characterization of the mathematical model from the article in terms of, e.g., integrality property and network flows. Explain your findings.

4. Heuristics[3]

   (a) Define a constructive heuristic for the model (1). Implement in Matlab or Julia and find a feasible solution to the instance in `as_mod.jl` and `as_dat_large.jl` (letting $d = 20$ and $T = 100$). Apply your algorithm also to the instance of (1) given by $T = 100$, $|\mathcal{N}| = 4$, $(c_{it})_{i\in\mathcal{N}} = (5, 6, 7, 9)$, $d_t = 10$, $t = 1, \ldots, T$, and $(T_i)_{i\in\mathcal{N}} = (3, 4, 5, 7)$.

   (b) Define a neigbourhood of a feasible solution to the model (1). A neighbourhood may be defined, e.g., with respect to all the variables or just the variables z_t, $t = 1, \ldots, T$. The search of the neighbourhood involves the solution of subproblems—a well chosen neighbourhood results in subproblems that are "easy" to solve (i.e., in polynomial time). Identify the subproblems resulting from your choice of neighbourhood and describe how these can be efficiently solved.

   (c) Define and implement a local search algorithm for the model (1) and use it to improve the respective solutions found in 4a.

[2] E. Gustavsson, M. Patriksson, A.-B. Strömberg, A. Wojciechowski, and M. Önnheim: *Preventive maintenance scheduling of multi-component systems with interval costs*, Computers & Industrial Engineering, vol. 76, pp. 390–400, 2014.
Open access, at `http://dx.doi.org/10.1016/j.cie.2014.02.009`

[3] You may carry out this exercise using the model from the article from 2014 as well.

(d) Draw illustrating schedules of the solutions found in 4a and 4c and compare with the optimal solution to this instance. Also, draw diagrams showing the objective value as a function of the number of iterations performed in the local search algorithm and CPU seconds, respectively.

5. Assume that it is required that the system (including all of its components) has a remaining life which is at least $r > 0$ time steps at the end of the planning period (i.e., at time $t = T$).[4]

   (a) Add and/or modify constraints to/in the model to accomplish this and solve the resulting model. Start by the model in `as_mod.jl` with data from `as_dat_large.jl` (with $d = 20$ and $T = 100$). Verify that the solution fulfills the requirement stated.

   (b) For five different relevant values of r (these values should be chosen such that the respective solutions become significantly different), compare the total cost for maintenance according to the schedule computed in 5a with that of the "original" (with $r = 0$) one. Comment on the number of maintenance occasions and the number of replaced components and compare with the corresponding numbers from the "original" model (with $r = 0$).

   (c) Which values of r are relevant for this study and why?

[4] You may carry out this exercise using the model from the article from 2014 as well.