Windfarm maintenance

Levente Fodor - mu1g7o

September 15, 2023

Contents

List of Figures

1 Windfarm problem setting

The following paper relies heavily on the course material of `Modelling and Optimization in practice` at University of Gyor that can be found at [2]. The implementation is done using `GNU Linear Programming Kit`, [1].
Several maintenance jobs specific to windfarm are to be carried out while minimizing costs. Costs are broken down to material and staff related components. Each maintenance job requires staff with different expertise and each job can be normal or severe maintenance that requires deeper knowledge from the staff to resolve. The staff also has an attribute called experience points or job expertise that can be junior/middle/senior the higher the level of experience the more XP is related but also higher staff costs are associated.

(a) Windfarm

(b) More windfarm.

Figure 1: Windfarms near Mosonmagyarovar, HU.

1.1 Maintenance jobs

Maintenance jobs are divided into 5 types based on which expertise is required from the staff. Each of the 5 maintenance types are further grouped based on severity. Level of severity determines the XPs required from the staff to accomplish such task. Currently only 2 severity levels have been set for each task types, "normal" stands for regular task and "severe" stands for more difficult operations with higher required XP.

1.2 Staff

Staff is divided into 4 types: electrician is taking care of wiring and electrical repairs, mechanic is tasked with assembling/disassembling stuffs, storage workers are keeping the warehouse tidy and in order while software engineers are responsible for programming controlling software and calibrating the elements. Staff is also described by level of expertise, there are 3 categories with different experience points: junior, middle, senior.

1.3 Sets

- set MaintenanceTypes {blades, gearbox, generator, sensors, wiring}
- set MaintenanceSeverity {normal, severe}
- set StaffTypes {electric, mechanic, storage, software}
- set StaffLevels {junior, middle, senior}

1.4 Parameters

- param main_req {MaintenanceTypes, MaintenanceSeverity}: cnt of maintenance required
- param main_req_st{MaintenanceTypes,StaffTypes}: cnt of staff required per maintenance type and staff category

- param main_req_xp{MaintenanceTypes,MaintenanceSeverity}: required XP for maintenance task based on severity
- param main_material_cost{MaintenanceTypes}: material cost of maintenance tasks
- param staff_level_xp{StaffLevels}: staff level XPs
- param staff_cost{StaffTypes,StaffLevels}: staff cost per type and level
- param main_burnout: weight based on severity of maintenance task
- param burnout_coef: total (severity weighted) tasks / staff cutoff

1.5 Variables

1.5.1 Variables used by solver

- var staff_to_hire{StaffTypes, StaffLevels}: cnt of staff needed as per type and level
- var quantity{MaintenanceTypes}: cnt maintenance tasks to be carried out

1.5.2 Redundant variables - improve readability

- var total_main_req_xp{MaintenanceTypes}: required XP points to carry out maintenance task
- var total_staff_xp{StaffTypes}: total available XP per staff types of hired personnel
- var total_staff{StaffTypes}: total cnt of hired personnel
- var total_staff_xp_task{MaintenanceTypes}: XP of hired personnel per category required for maintenance task
- var weighted_maintenance_tasks{MaintenanceTypes}: nr of maintenance tasks weighted by burnout factor
- var total_req_wgt_staff{StaffTypes}: total cnt of required staff for maintenance tasks weighted by severity

1.6 Conditions

- *XP condition:* sufficient XP points to carry out upcoming maintenance tasks based on severity. It is only required to have enough XPs to carry out at least one of all types of maintenance task (based on severity), e.g. maintenance task "blades" normally requires 50 XPs whereas severe requires 125 XPs but when no severe maintenance task is scheduled for "blades" (main_req parameter is set to 0) then the condition prescribes 50 XPs for this maintenance task (normal severity level). It does not matter how many maintenance jobs are expected, the hired staff has to have enough XPs to do all types of jobs. E.g. if normal maintenance task "blades" is expected 12 times and no severe "blades" task required, then XPs of hired personnel has to have at least 50 XPs (normal "blade" maintenance XP) and not 12 times 50 XPs. It will just take them more time to do all 12 normal "blades" job.
- *Staff condition:* number of staff for a given type (no matter if junior/middle/senior) covers the required number of staff for each maintenance task. E.g. in order to be able carry out maintenance task "blades", main_req_st{MaintenanceType,StaffType} prescribes 3 mechanics and 4 storage personnel so at least 3 mechanics and 4 storage workers have to be hired.
- *Minimum maintenance:* maintenance jobs defined in main_req are carried out.
- *Burnout indicator:* staff is not overloaded meaning that the severity weighted number of maintenance tasks per staff member has to be lower than the parameter *burnout_coef*.

1.7 Objectives

Minimize total costs given by the aggregated sum of main_material_cost and staff_cost given the above conditions.

2 Windfarm problem implementation

2.1 Summary

This implementation of the windfarm maintenance problem is done using GLPK/GMPL (GNU Linear Programming Kit and Modelling language GNU MathProg) [1]. The program is broken down to 3 input and 2 output files:

- Input files
  - *windfarm.mod*: model file containing the core of the modelling problem. Declares sets, parameters, conditions and the objective of the modelling exercise. Also, the file includes *printf* statements that generates more readable output than the standard built-in one in *GLPK*. This customized output is available in the *report.txt* file.
  - *wfmaintenance.dat*: data file with initialization of sets and parameters related to maintenance jobs. Additionally, the parameter matrix *main_req_st* that links maintenance tasks and staff related parameters is also initialized here. Following parameters are initialized here:
    - set MaintenanceTypes
    - set MaintenanceSeverity
    - param main_req
    - param main_req_xp
    - param main_req_st
    - param main_burnout
    - param burnout_coef
    - param main_material_cost
  - *wfstaff.dat*: data file initializing staff related sets and parameters.
    - set StaffTypes
    - set StaffLevels
    - param staff_level_xp
    - param staff_cost
- Output files
  - *output.txt*: standard output file generated by the *glpsol* command with the *-o* parameter. The file shows the values of all variables and the values used for resolving the conditions specified in the .mod file.
  - *report.txt*: customized output generated by the *printf* statements at the end of the *windfarm.mod* file. Contentwise same as *output.txt*.

2.2 Model setup

2.2.1 Conditions and variables

The condition section of the modelling file is broken down to 4 main sections, XP conditions, Staff conditions, Minimum maintenance and Burnout indicators respectively. The XP condition takes care of that the hired staff have at least as much XP points that is necessary to be able to carry out the required maintenance tasks. The following variables are calculated within this section:

XP conditions
Firstly, redundant variables:

$$\underset{mt\times 1}{\text{total\_main\_req\_xp}} = \begin{cases} \underset{mt\times ms}{\text{main\_req\_xp[mt,"severe"]}}, \text{ if } \underset{mt\times ms}{\text{main\_req\_xp[mt,"severe"]}} > 0 \\ \underset{mt\times ms}{\text{main\_req\_xp[mt,"normal"]}}, \text{ otherwise} \end{cases} \tag{1}$$

$$\underset{st\times 1}{\text{total\_staff\_xp}} = \underset{st\times sl}{\text{staff\_to\_hire}} \times \underset{sl\times 1}{\text{staff\_level\_xp}} \tag{2}$$

$$\underset{mt\times 1}{\text{total\_staff\_xp\_task}} = \begin{cases} \sum_{st} \underset{st\times 1}{\text{total\_staff\_xp[st]}}, \text{ if } \underset{mt\times st}{\text{main\_req\_st[mt,st]}} > 0 \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

Finally, the required condition is

$$\underset{mt\times 1}{\text{total\_staff\_xp\_task}} >= \underset{mt\times 1}{\text{total\_main\_req\_xp}} \tag{4}$$

Staff conditions
Again, introduce (this time only one) redundant variable:

$$\underset{st\times 1}{\text{total\_staff}} = \sum_{sl} \underset{st\times sl}{\text{staff\_to\_hire[st,sl]}} \tag{5}$$

The staff condition is

$$\underset{st\times 1}{\text{total\_staff}} >= \underset{mt\times st}{\text{main\_req\_st}^T\text{[mt,st]}} \tag{6}$$

Minimum maintenance
No redundant variables this time:

$$\underset{mt\times 1}{\text{quantity}} >= \sum_{ms} \underset{mt\times ms}{\text{main\_req[mt,ms]}} \tag{7}$$

Burnout condition
Redundant variables:

$$\underset{mt\times 1}{\text{weighted\_maintenance\_tasks}} = \underset{mt\times ms}{\text{main\_req[mt,ms]}} \times \underset{ms\times 1}{\text{main\_burnout[ms]}} \tag{8}$$

$$\underset{1\times st}{\text{total\_req\_wgt\_staff}} = \underset{mt\times 1}{\text{weighted\_maintenance\_tasks}^T\text{[mt]}} \times \underset{mt\times st}{\text{main\_req\_st[mt,st]}} \tag{9}$$

The required burnout condition is:

$$\underset{st\times 1}{\text{total\_req\_wgt\_staff[st]}} <= \underset{st\times 1}{\text{total\_staff[st]}} * \text{burnout\_coef} \tag{10}$$

where
mt is in MaintenanceTypes
ms is in MaintenanceSeverity
st is in StaffTypes
st is in StaffLevels
sets.

2.2.2 Objective

The objective function of the modelling exercise is minimizing total costs obtained as the sum of material and staff related costs while fulfilling required maintenance tasks given the conditions. Total staff cost is obtained by multiplying matrices *staff_to_hire* and *staff_cost* and then taking the trace of the resulting matrix:

$$\text{TotalStaffCost} = \text{tr}\left(\underset{(st\times sl)}{\text{staff\_to\_hire}} \times \underset{(st\times sl)}{\text{staff\_cost}^T}\right)$$

Total material cost can be obtained by simply taking the sumproduct of vectors *main_material_cost* and *quantity*:

$$\text{TotalMaterialCost} = \sum_{mt} \text{main\_maintenance\_cost[mt]} * \text{quantity[mt]}$$

Objective to be minimized:

$$\text{minimize TotalCosts} = \text{TotalStaffCost} + \text{TotalMaterialCost} \tag{11}$$

2.3 Sample input/output

2.3.1 Parameter main_req

The parameter contains the required number of maintenance jobs (rows) based on severity (columns). The required number of maintenance jobs per types is given by the row sums (summed over severity level). The row sums are used in the condition Minimum maintenance.

Maintenance Type	Maintenance Severity	
	normal	severe
blades	12	5
gearbox	13	3
generator	6	10
sensors	8	1
wiring	9	0

Table 1: main_req

2.3.2 Parameter main_req_xp

The parameter contains the required experience points for the maintenance jobs (rows) based on severity (columns). In case the value for the "severe" task in *main_req* is greater than zero, the experience points defined in this parameter matrix is taken from the "severe" column and from the "normal" otherwise (see XP conditions).

Maintenance Type	Maintenance Severity	
	normal	severe
blades	50	125
gearbox	125	150
generator	113	230
sensors	110	150
wiring	30	50

Table 2: main_req_xp

2.3.3 Parameter main_req_st

The parameter holds the number of personnel required to do the specific maintenance job. E.g. the "blades" task requires 3 mechanics and 4 storage personnel to move the blades from the storage to the windfarm site and fix them.

Maintenance Type	Staff Type			
	electric	mechanic	storage	software
blades	0	3	4	0
gearbox	3	4	0	1
generator	4	3	0	1
sensors	5	3	1	3
wiring	3	0	3	2

Table 3: main_req_st

2.3.4 Parameter main_burnout and burnout_coef

The parameter tells how exhausting the task is for the staff based on severity. The higher the value, the more the task weighs when calculating the number of tasks per staff member. The latter is represented by the parameter *burnout_coef* and is set to **15** meaning that no more than 15 tasks can be assigned to 1 staff member.

Maintenance Severity	Value
normal	.75
severe	1.3

Table 4: main_burnout

2.3.5 Parameter main_material_cost

The parameter contains the cost of material associated with the maintenance tasks.

Maintenance Type	Material Cost
blades	100
gearbox	150
generator	120
sensors	50
wiring	70

Table 5: main_material_cost

2.3.6 Parameter staff_level_xp

The parameter contains the experience points associated to level of the staff. The higher the level, the more experience point the person has (and also more expensive).

Staff Level	XP
junior	2
middle	15
senior	40

Table 6: staff_level_xp

2.3.7 Parameter staff_cost

The parameter holds the cost of hiring staff of given type and experience level.

Staff Type	Staff Level		
	junior	middle	senior
electric	10	15	40
mechanic	8	18	30
storage	12	15	20
software	10	40	50

Table 7: staff_cost

2.3.8 Generated output

The section shows the generated output based on the above input and model setup. The output can be found in the **output.txt** and also in the **report.txt** files. The latter is scripted in the *windfarm.mod* and provides a more readable format of the output. The first printed section in the *report.txt* file is the *total_main_req_xp* which is just the required experience points per maintenance tasks depending on whether only normal or also severe occurences of the given tasks are expected in the problem setting (more precisely in the *main_req* parameter).

```
----------------------------------------
total_main_req_xp
blades 125
gearbox 150
generator 230
sensors 150
wiring 30
```

Secondly, the experience points of the staff per maintenance tasks are printed. The previous vector and the one below are to be compared in (4) so that *total_staff_xp_task* is expected to be greater or equal than *total_main_req_xp* for all maintenance types.

```
----------------------------------------
total_staff_xp_task
blades 125
gearbox 238
generator 238
sensors 263
wiring 163
```

Next, take a look at *total_staff* to make sure we have greater or equal number of staff per staff types than as required in parameter *main_req_st* (6).

```
----------------------------------------
total_staff
electric 12
mechanic 12
storage 6
software 5
```

Next up is the burnout constraint (8) ensuring that the severity weighted count of maintenance jobs per employee does not exceed the parameter $burnout\_coef = 15$ on staff type level.

```
----------------------------------------
total_req_wgt_staff over total_staff
electric 13.975000
mechanic 14.625000
storage  14.925000
software 13.310000
```

Finally, we would like to see how many employees we need to hire with what level of experience and of course what cost impact it generates. To obtain this, the variable *staff_to_hire* and the product of this with parameter *staff_cost* are printed.

```
----------------------------------------
staff_to_hire and staff cost ===== 425
electric   ========== 160
junior     4......... 40
middle     8......... 120
senior     0......... 0

mechanic   ========== 140
junior     10........ 80
middle     0......... 0
senior     2......... 60

storage    ========== 75
junior     5......... 60
middle     1......... 15
senior     0......... 0

software   ========== 50
junior     5......... 50
middle     0......... 0
senior     0......... 0
```

Read the above table as follows: e.g. staff type electric requires 4 juniors and 8 middle level employees, no senior level staff is required. The cost of hiring 4 juniors is 40 whereas the mid-level colleagues cost 120, totalling 160. Summing all the categories we obtain total staff cost of 425.

```
----------------------------------------
main_material_cost ===== 7163
blades......1700
gearbox......2400
generator......1920
sensors......450
wiring......693
----------------------------------------
total costs = SUM material costs + SUM staff costs
7588 = 7163 + 425
```

Lastly, material costs are also displayed and the sum of material and staff cost to obtain the total costs on firm level.

2.4 Effect of parameters

The last section of this paper is dedicated to impact analysis of some of the parameters on modelling outputs. Different scenarios are defined by means of different input parameters and the resulting *staff_to_hire[st,sl]* is shown. The present modelling setup is focusing mainly on the number of employees per staff types that is in variable *staff_to_hire[st,sl]*. The other changing variable, the *quantity[mt]* is only affecting the final costs through the material costs associated with the maintenance jobs, hence the main focus of this section is the staff related variable. The model might be further enhanced with additional constraints giving more importance to the *quantity[mt]* variable.

By observing the XP conditions section, it can be noticed that the first element having an effect on variable *staff_to_hire* is the parameter *staff_level_xp* defining the experience points associated to different levels of the employee. According to the conditions, the staff to be hired has to have enough experience points to be able to do all maintenance jobs. The required experience points

for a maintenance job type is determined by *main_req_xp* (number of XPs per job types based on severity) and parameter *main_req* that tells if a job type with "severe" category is expected or not.

Following the XPs, the Staff conditions section hints that *main_req_st* also affects the hiring aspect of the results, in other words, we need different staff composition if the maintenance jobs require different number of personnel from different job types. E.g. if job type "blades" would require electricians as well, or different number of mechanics, we would end up with different outcome.

Finally, the burnout effect (through the related *burnout_coef* and *main_burnout*) is also taking impact. It is relatively easy to see that in case of the coefficient a higher value relaxes the condition (1 employee can take on more tasks without burning out) whereas changing the weights has opposite effect (higher values increases the required number of employees). As the burnout effect is the most straightforward - as it affects all job and staff types the same way - the analysis section is reduced to the effects of parameters setting the XPs and staff costs. In this aspect, it is decisive how much more a senior is paid compared to a junior or middle level staff and also how many employees are required from the different staff types to complete each maintenance jobs.

Based on the above overview, the following parameters will be changed and analysed:

- main_req_st
- main_req_xp
- staff_level_xp
- staff_cost

2.4.1 Δ main_req_st

Table 3 shows the required personnel to carry out the different type of tasks. Let's suppose, we start to install a new type of gearbox that requires only 1 electrician and 2 mechanics (instead of 3 and 4 respectively).

```
----------------------------------------
staff_to_hire and staff cost ===== 392
electric    ========== 140
junior     2......... 20
middle     8......... 120
senior     0......... 0

mechanic    ========== 124
junior     8......... 64
middle     0......... 0
senior     2......... 60

storage    ========== 78
junior     4......... 48
middle     2......... 30
senior     0......... 0

software    ========== 50
junior     5......... 50
middle     0......... 0
senior     0......... 0
```

As expected, the staff cost is down from 425 to 392, and ceteris paribus we need only 2 junior electricians and 8 junior mechanics.

2.4.2 Δ main_req_xp

Table 2 shows the required experience points per maintenance jobs in job severity breakdown. Assume now, that a severe "blades" job requires only 75 experience points instead of 125 and a normal "wiring" requires 50 (instead of 30) just like the severe version.

```
-----------------------------------------
staff_to_hire and staff cost ===== 418
electric   ========== 175
junior     1......... 10
middle     11........ 165
senior     0......... 0

mechanic   ========== 118
junior     11........ 88
middle     0......... 0
senior     1......... 30

storage    ========== 75
junior     5......... 60
middle     1......... 15
senior     0......... 0

software   ========== 50
junior     5......... 50
middle     0......... 0
senior     0......... 0
```

Due to the aforementioned changes - ceteris paribus - number of junior electricians are down to 1 and middle level electricians increases to 11 (4 and 8 original values), electrician subgroup in staff cost is up to 175 (from 160). Additionally, 11 junior mechanics are required and only 1 senior (instead of 10 and 2), driving the staff cost on mechanics from 140 down to 118. The total staff cost change is therefore -7 (from 425 to 418).

2.4.3 Δ staff_level_xp

Consider this time table 6 with the experience points associated with staff levels. Assume an increase of experience points in of middle level employees from 15 to 30.

```
-----------------------------------------
staff_to_hire and staff cost ===== 385
electric   ========== 155
junior     5......... 50
middle     7......... 105
senior     0......... 0

mechanic   ========== 96
junior     12........ 96
middle     0......... 0
senior     0......... 0

storage    ========== 84
junior     2......... 24
middle     4......... 60
senior     0......... 0

software   ========== 50
junior     5......... 50
middle     0......... 0
senior     0......... 0
```

The ceteris paribus increase of XPs of the middle level staff causes that no senior employees are needed to fulfill the jobs and also pushes the staff cost down to 385 from 425. However, this is not the most realistic scenario as the higher XPs of middle level staff would also increase their (expected) salaries which eventually could balance out their advantage.

2.4.4 Δ staff_cost

Finally, we managed to hire some senior electricians for much lower salary, the corresponding value in table 7 is down from 40 to 20.

```
----------------------------------------
staff_to_hire and staff cost ===== 407
electric   ========== 170
junior     6......... 60
middle     2......... 30
senior     4......... 80

mechanic   ========== 96
junior     12........ 96
middle     0......... 0
senior     0......... 0

storage    ========== 91
junior     3......... 36
middle     1......... 15
senior     2......... 40

software   ========== 50
junior     5......... 50
middle     0......... 0
senior     0......... 0
```

This single parameter causes changes in almost all staff types except for the software engineers. When senior electricians earn 20 instead of 40, this makes the company to hire 4 senior electricians, 2 middle level and 6 juniors instead of 0, 8, 4 respectively in the baseline scenario. Also, the senior electricians' salary correlates with the mechanics where the company no longer looks for hiring 2 seniors, instead would replace them with 2 juniors. In the warehouse however, 2 juniors are let go and 2 seniors are welcome. The total staff cost is down to 407 from 425.

A Source code for modelling

```
# set model windfarm
set MaintenanceTypes;
set MaintenanceSeverity;
set StaffTypes;
set StaffLevels;

# cnt of maintenance required
param main_req{mt in MaintenanceTypes, ms in MaintenanceSeverity};

# cnt of staff required per maintenance type
param main_req_st{mt in MaintenanceTypes, st in StaffTypes};

# required xp for maintenance task
param main_req_xp{mt in MaintenanceTypes, ms in MaintenanceSeverity};

# maintenance cost
param main_material_cost{MaintenanceTypes};

param staff_level_xp{StaffLevels};
param staff_cost{st in StaffTypes, sl in StaffLevels};

# weights to indicating staff's exhaustion due to work on task with
# given severity
param main_burnout{MaintenanceSeverity};

# bearable level of staff's exhaustion
param burnout_coef;

## variables
# required XP points to carry out maintenance task
var total_main_req_xp{MaintenanceTypes};

# total available XP per staff types of hired personnel
var total_staff_xp{StaffTypes};

# total cnt of hired personnel
var total_staff{StaffTypes};

# XP of hired personnel per staff category
var total_staff_xp_task{MaintenanceTypes};

# nr of maintenance tasks weighted by burnout factor
var weighted_maintenance_tasks{MaintenanceTypes};

# total cnt of required staff for severity weighted maintenance tasks
var total_req_wgt_staff{StaffTypes};

var staff_to_hire{StaffTypes, StaffLevels} >=0, integer;
var quantity{MaintenanceTypes} >= 0;
```

```
## conditions
# XP: sufficient XP points to carry out upcoming maintenance tasks
# based on severity
s.t. RequiredXP mt in MaintenanceTypes :
    total_main_req_xp mt = (if main_req mt,"severe" <> 0 then
                                main_req_xp mt,"severe"
                                else main_req_xp mt,"normal");

s.t. AvailableXP st in StaffTypes :
    total_staff_xp st = sum sl in StaffLevels staff_to_hire st,sl *
                            staff_level_xp sl ;

s.t. AvailableXP_task mt in MaintenanceTypes :
    total_staff_xp_task mt = sum st in StaffTypes
                                if main_req_st mt,st <> 0 then
                                total_staff_xp st else 0 ;

s.t. AvailableXP_ge_RequiredXP mt in MaintenanceTypes :
    total_staff_xp_task mt >= total_main_req_xp mt ;

# staff: enough personnel to cover staff requirements per maintenance
# tasks
s.t. AvailableStaff st in StaffTypes :
    total_staff st = sum sl in StaffLevels staff_to_hire st,sl ;

s.t. AvailableStaff_ge_RequiredStaff mt in MaintenanceTypes,
st in StaffTypes :
    total_staff st >= main_req_st mt, st ;

# minimum maintenance
s.t. RequiredMaintenanceDone mt in MaintenanceTypes :
    quantity mt >= sum ms in MaintenanceSeverity main_req mt,ms ;

# burnout indicators
s.t. StaffNoBurnout mt in MaintenanceTypes :
    weighted_maintenance_tasks mt = sum ms in MaintenanceSeverity
        main_req mt,ms * main_burnout ms ;

s.t. TotalReqWgtStaff st in StaffTypes :
    total_req_wgt_staff st = sum mt in MaintenanceTypes
        weighted_maintenance_tasks mt * main_req_st mt,st ;
    #total_req_wgt_staff[st] = sum{mt in MaintenanceTypes}
        # weighted_maintenance_tasks[mt] *
        # (if main_req_st[mt,st] == 0 then 0 else 1);

# burnout
s.t. BurnOutNotAllowed st in StaffTypes :
    total_req_wgt_staff st <= total_staff st * burnout_coef;

## Objective
minimize TotalCosts:
    sum st in StaffTypes, sl in StaffLevels staff_to_hire st,sl *
        staff_cost st,sl +
    sum mt in MaintenanceTypes ( main_material_cost mt ) *
        quantity mt ;

solve;
```

B Source code of maintenance data file

```
# set data maintenance
data;

set MaintenanceTypes :=
    blades
    gearbox
    generator
    sensors
    wiring;

set MaintenanceSeverity :=
    normal
    severe;

param main_req :
            normal                  severe :=
    blades 12 5
    gearbox 13 3
    generator 6 10
    sensors 8 1
    wiring 9 0;

param main_req_xp :
            normal severe :=
    blades 50 125
    gearbox 125 150
    generator 113 230
    sensors 110 150
    wiring 30 50;

param main_req_st :
            electric mechanic storage software :=
    blades 0 3 4 0
    gearbox 3 4 0 1
    generator 4 3 0 1
    sensors 5 3 1 3
    wiring 3 0 3 2;

param main_burnout :=
    normal .75
    severe 1.3;

param burnout_coef := 15 ;

param main_material_cost :=
    blades 100
    gearbox 150
    generator 120
    sensors 50
    wiring 77;

end;
```

C Source code of staff data file

```
# set data staff
data;

set StaffTypes :=
    electric
    mechanic
    storage
    software;

set StaffLevels :=
    junior
    middle
    senior;

param staff_level_xp :=
    junior 2
    middle 15
    senior 40;

param staff_cost :
            junior middle senior :=
    electric 10 15 40
    mechanic 8 18 30
    storage 12 15 20
    software 10 40 50;

end;
```

D Output

```
Problem:    windfarm
Rows:       62
Columns:    44 (12 integer, 0 binary)
Non-zeros:  141
Status:     INTEGER OPTIMAL
Objective:  TotalCosts = 7588 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 RequiredXP blades
                                 125           125             =
     2 RequiredXP gearbox
                                 150           150             =
     3 RequiredXP generator
                                 230           230             =
     4 RequiredXP sensors
                                 150           150             =
     5 RequiredXP wiring
                                  30            30             =
     6 AvailableXP electric
                                   0            -0             =
     7 AvailableXP mechanic
                                   0            -0             =
     8 AvailableXP storage
                                   0            -0             =
     9 AvailableXP software
                                   0            -0             =
    10 AvailableXP_task blades
                                   0            -0             =
    11 AvailableXP_task gearbox
                                   0            -0             =
    12 AvailableXP_task generator
                                   0            -0             =
    13 AvailableXP_task sensors
                                   0            -0             =
    14 AvailableXP_task wiring
                                   0            -0             =
    15 AvailableXP_ge_RequiredXP blades
                                   0            -0
    16 AvailableXP_ge_RequiredXP gearbox
                                  88            -0
    17 AvailableXP_ge_RequiredXP generator
                                   8            -0
    18 AvailableXP_ge_RequiredXP sensors
                                 113            -0
    19 AvailableXP_ge_RequiredXP wiring
                                 133            -0
```

```
20 AvailableStaff electric
0           -0           =
21 AvailableStaff mechanic
       0            -0            =
22 AvailableStaff storage
       0            -0            =
23 AvailableStaff software
       0            -0            =
24 AvailableStaff_ge_RequiredStaff blades,electric
    12           -0
25 AvailableStaff_ge_RequiredStaff blades,mechanic
    12            3
26 AvailableStaff_ge_RequiredStaff blades,storage
       6            4
27 AvailableStaff_ge_RequiredStaff blades,software
       5            -0
28 AvailableStaff_ge_RequiredStaff gearbox,electric
    12            3
29 AvailableStaff_ge_RequiredStaff gearbox,mechanic
    12            4
30 AvailableStaff_ge_RequiredStaff gearbox,storage
       6            -0
31 AvailableStaff_ge_RequiredStaff gearbox,software
       5            1
32 AvailableStaff_ge_RequiredStaff generator,electric
    12            4
33 AvailableStaff_ge_RequiredStaff generator,mechanic
    12            3
34 AvailableStaff_ge_RequiredStaff generator,storage
       6            -0
35 AvailableStaff_ge_RequiredStaff generator,software
       5            1
36 AvailableStaff_ge_RequiredStaff sensors,electric
    12            5
37 AvailableStaff_ge_RequiredStaff sensors,mechanic
    12            3
38 AvailableStaff_ge_RequiredStaff sensors,storage
       6            1
39 AvailableStaff_ge_RequiredStaff sensors,software
       5            3
```

```
40 AvailableStaff_ge_RequiredStaff wiring,electric
12            3
41 AvailableStaff_ge_RequiredStaff wiring,mechanic
      12            -0
42 AvailableStaff_ge_RequiredStaff wiring,storage
      6             3
43 AvailableStaff_ge_RequiredStaff wiring,software
      5             2
44 RequiredMaintenanceDone blades
      17            17
45 RequiredMaintenanceDone gearbox
      16            16
46 RequiredMaintenanceDone generator
      16            16
47 RequiredMaintenanceDone sensors
      9             9
48 RequiredMaintenanceDone wiring
      9             9
49 StaffNoBurnout blades
    15.5          15.5            =
50 StaffNoBurnout gearbox
    13.65         13.65             =
51 StaffNoBurnout generator
    17.5          17.5            =
52 StaffNoBurnout sensors
    7.3           7.3             =
53 StaffNoBurnout wiring
    6.75          6.75            =
54 TotalReqWgtStaff electric
      0             -0              =
55 TotalReqWgtStaff mechanic
      0             -0              =
56 TotalReqWgtStaff storage
      0             -0              =
57 TotalReqWgtStaff software
      0             -0              =
58 BurnOutNotAllowed electric
    -12.3                          -0
59 BurnOutNotAllowed mechanic
    -4.5                         -0
```

```
      60 BurnOutNotAllowed storage
      -0.45                          -0
      61 BurnOutNotAllowed software
             -8.45                           -0
      62 TotalCosts                7588

      No. Column name        Activity     Lower bound   Upper bound
      ------ ------------    ------------- ------------- -------------
      1 total_main_req_xp blades
             125
      2 total_main_req_xp gearbox
             150
      3 total_main_req_xp generator
             230
      4 total_main_req_xp sensors
             150
      5 total_main_req_xp wiring
             30
      6 total_staff_xp electric
             128
      7 total_staff_xp mechanic
             100
      8 total_staff_xp storage
             25
      9 total_staff_xp software
             10
      10 total_staff electric
             12
      11 total_staff mechanic
             12
      12 total_staff storage
                 6
      13 total_staff software
                 5
      14 total_staff_xp_task blades
             125
      15 total_staff_xp_task gearbox
             238
      16 total_staff_xp_task generator
             238
      17 total_staff_xp_task sensors
             263
      18 total_staff_xp_task wiring
             163
      19 weighted_maintenance_tasks blades
             15.5
      20 weighted_maintenance_tasks gearbox
             13.65
```

```
21 weighted_maintenance_tasks generator
17.5
22 weighted_maintenance_tasks sensors
      7.3
23 weighted_maintenance_tasks wiring
      6.75
24 total_req_wgt_staff electric
    167.7
25 total_req_wgt_staff mechanic
    175.5
26 total_req_wgt_staff storage
    89.55
27 total_req_wgt_staff software
    66.55
28 staff_to_hire electric,junior
*             4             0
29 staff_to_hire electric,middle
*             8             0
30 staff_to_hire electric,senior
*             0             0
31 staff_to_hire mechanic,junior
*            10             0
32 staff_to_hire mechanic,middle
*             0             0
33 staff_to_hire mechanic,senior
*             2             0
34 staff_to_hire storage,junior
*             5             0
35 staff_to_hire storage,middle
*             1             0
36 staff_to_hire storage,senior
*             0             0
37 staff_to_hire software,junior
*             5             0
38 staff_to_hire software,middle
*             0             0
39 staff_to_hire software,senior
*             0             0
40 quantity blades
      17            0
```

```
41 quantity gearbox
16              0
42 quantity generator
        16              0
43 quantity sensors
        9               0
44 quantity wiring
        9               0

Integer feasibility conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
max.rel.err = 0.00e+00 on row 0
High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
max.rel.err = 0.00e+00 on row 0
High quality

End of output
```

References

[1] *GNU Linear Programming Kit.* URL: https://www.gnu.org/software/glpk/. (accessed: 01.09.2023).

[2] Mate Hegyhati. *Introduction into the modeling and optimization of linear systems.* URL: https://hegyhati.github.io/IMOLS/. (accessed: 01.09.2023).