

# SolveDB Exercises

Download and install the VM image with SolveDB and database for exercises: [https://drive.google.com/file/d/14g8CuwSY\\_R9Kw93NSSFv-Jj-XGPTdU4T/view?usp=sharing](https://drive.google.com/file/d/14g8CuwSY_R9Kw93NSSFv-Jj-XGPTdU4T/view?usp=sharing)

Check the “cheat-sheet” for quick start with SolveDB: [http://www.daisy.aau.dk/?page\\_id=1388](http://www.daisy.aau.dk/?page_id=1388)

## Warm-Up Exercises

### Problem 1: Knapsack optimization problem

Given weights and values of 4 items, (10.0, 5.0), (9.0, 4.5), (1.5, 2.0), (7.0, 3.0), put these items in the knapsack of capacity  $W = 15$  to get the maximum total value in the knapsack.

**Task 1.1:** Create a table in SolveDB for item weights and values

**Task 1.2:** Populate the table with the item weights and values

**Task 1.3:** Formulate and solve this Knapsack problem in SolveDB

**Task 1.4:** Solve this problem as the LP problem and compare the results

Given item weights and values in the table  $n\_items$ , put these items in:

**Task 1.5:** a single knapsack of capacity  $W = 150$ . Use the GLPK solver (“solverlp.auto”).

**Task 1.6:** a single knapsack of capacity  $W = 150$ . Use the CBC solver (“solverlp.cbc”).

**Task 1.7:**  $N$  knapsacks of capacity  $W = 150$  according to the “knr” column.

### Problem 2: Simple LP Problem

A small business enterprise makes dresses and trousers. To make a dress requires  $1/2$  hour of cutting and 20 minutes of stitching. To make a trousers requires 15 minutes of cutting and  $1/2$  hour of stitching. The profit on a dress is 40 DKK and on a pair of trousers 50 DKK. The business operates for a maximum of 8 hours a day. Determine how many dresses and trousers should be made to maximize profit.

**Task 2.1:** Translate from this verbal to a symbolic form.

**Task 2.2 [Optional]:** Solve this LP graphically or any other optimization tool.

**Task 2.2:** Formulate and solve this problem in SolveDB.

**Task 2.3:** Update the SolveDB problem, so data comes from database tables.

### Problem 3: Sudoku problem

Consider the Sudoku puzzle: a puzzle setter provides a partially filled  $9 \times 9$  matrix, and the goal is to complete filling the matrix with the digits 1 to 9 so that each column, each row, and each of the nine  $3 \times 3$  sub-matrices have no duplicated digits (see the table below).

**Task 3.1:** Formulate and solve the Sudoku puzzle using the given Sudoku values in the table *sudoku*.

**Task 3.2:** Solve the Sudoku problem using the specialized solver “sudokusolver”.

Hint: Check the *cheat-sheet* on how to formulate “distinct values” constraint.

Table 1: A Sudoku matrix with given and solution digits

<u>2</u>	9	6	4	<u>8</u>	3	<u>7</u>	1	5
8	4	7	5	<u>1</u>	2	3	<u>9</u>	6
1	<u>3</u>	5	<u>6</u>	7	<u>9</u>	4	<u>2</u>	8
6	7	8	3	9	<u>1</u>	<u>5</u>	4	2
<u>4</u>	1	2	8	6	5	9	7	<u>3</u>
3	5	<u>9</u>	<u>7</u>	2	4	8	6	1
9	<u>6</u>	3	<u>2</u>	5	7	1	<u>8</u>	4
5	<u>2</u>	1	9	<u>4</u>	8	6	3	7
7	8	<u>4</u>	1	<u>3</u>	6	2	5	<u>9</u>

## Problem 4: NLP curve fitting problem

Given a table  $points(id,x,y)$  with a number of data points, find the best fit of the function  $f(x) \mapsto a \cdot \sin(x) + b$  where  $a$  and  $b$  are the unknown (decision) parameters:

**Task 4.1:** Use the “solversw” and least-squares in SolveDB.

**Task 4.2:** Use the “solverlp” and the least absolute deviations. Compare the results, solving time, and solving quality.

Hint: See the cheat-sheet on how to use auxiliary variables for the  $abs$ . For further details, read: [https://en.wikipedia.org/wiki/Least\\_absolute\\_deviations](https://en.wikipedia.org/wiki/Least_absolute_deviations)

## Problem 5: “How To?” queries

A shipping company keeps records of open order requests in the (TPC-H inspired) database. Each order consists of several line items, stored in the relation  $lineitems10000(orderkey,partkey,supplierkey,quant)$  with the key  $(orderkey,partkey,supplierkey)$ : a record means that the order  $orderkey$  contains a line item the part  $partkey$  will be delivered from the supplier  $supplierkey$ .

Suppose that, due to a change in corporate policies, the company decides to limit the total quantity per order and supplier to at most 50.

**Task 5.1:** Produce another table (or view) such that total quantity per order is no more than 50.

**Task 5.2:** Delete line items in the table such that the total quantity per order is no more than 50.

**Task 5.3** (Optional): For Task 5.1 or 5.2, write a standard SQL query and then compare it with the one using SOLVESELECT.

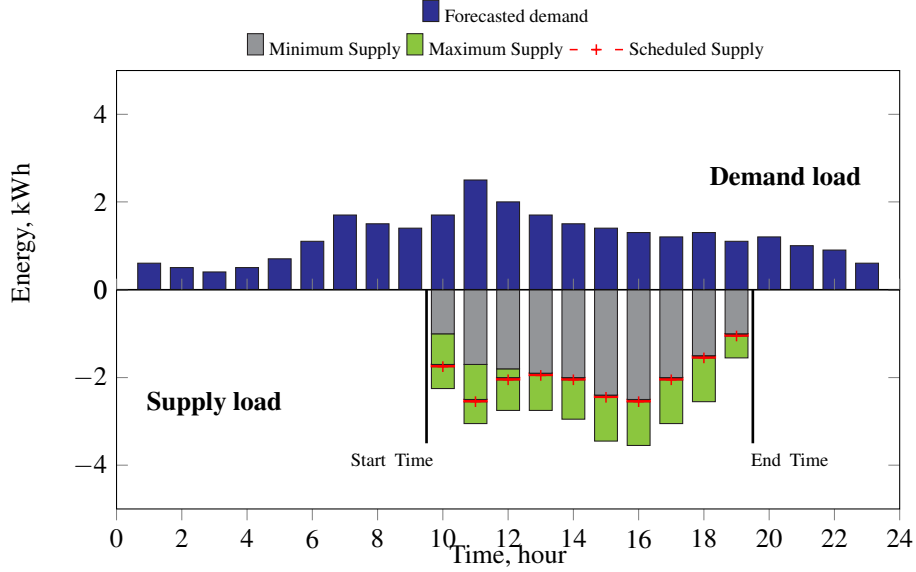
## Energy Planning Scenarios

### Problem 6: Demand and Supply Balancing

An electricity supplier supplies electricity to (inflexible) residential electricity consumers by activating a number of (flexible) production units. Before the day of operation, the supplier needs to plan the activation of the production units to be able to satisfy the demand according to the prognosis of the residential consumption, i.e., the expected hourly consumption and production amounts should be equal for each of the 24 hours. The total residential demand historical data is stored in the database table  $hist\_load(time,load)$ . The production unit flexibility data is stored in the table  $hist\_prod(fid,tid,el,eh,price)$ , where  $fid$  is a flexible production units Id;  $tid$  - a planning time interval Id (= hour within the day);  $el$  and  $eh$  - minimum and maximum energy amounts the unit can (should) deliver, respectively;  $price$  - the cost of the production (price per kWh).

**Task 6.1:** Use historical demand data to forecast loads in the next 24 hours (2017-07-01 – 2017-07-02). Use the pre-installed “forecastingsolver” in SolveDB. More proficient students may define SolveDB views

Figure 1: Forecasted demand and flexible supply loads



for forecasting loads using (multiple) linear regression (LR) by using the features (explanatory variables) given by the view *hist\_features*. Hint: use least absolute deviations to tune the parameters of the LR model:

$$load_h = pmon_h \cdot mon_h + ptue_h \cdot tue_h + \dots + psat_h \cdot sat_h + \varepsilon \quad (1)$$

where  $load_h$  is the expected load variable,  $mon_h, tue_h, \dots, sat_h$  are given explanatory variables, and  $pmon_h, ptue_h, \dots, psat_h$  are parameters (to be tuned), and  $\varepsilon$  is the error term (to be tuned), and  $h$  is the forecasting period 0..23 (hour).

Hint: Check the cheat-sheet on how to minimize absolute deviations.

**Task 6.2:** Defined and solve the cost optimization problem to allocate production loads within their constraints so that the forecasted demand loads are matched at every hour. Handle infeasibility (if arises due to too tight constraints) by introducing variables for error (mismatch) between demand and supply. Hint: model “abs” using the technique explained in the cheat-sheet or [https://download.aimms.com/aimms/download/manuals/AIMMS30M\\_LinearProgrammingTricks.pdf](https://download.aimms.com/aimms/download/manuals/AIMMS30M_LinearProgrammingTricks.pdf)

## Problem 7: Heat-pump power optimization

A residential electricity consumer operates an electrical heat-pump (HP) under the dynamic hourly tariff. The consumer tolerates the 20 – 24°C temperature comfort range, and so there is an option to pre-heat the house before electricity becomes expensive and then cool down the house when price becomes high. All historical data is stored in the table *nist(ts, intemp, outtemp, hppower, price)*, where *ts* is a time stamp; *intemp* - room temperature inside the house; *outtemp* - outside (ambient) temperature; *hppower* - average power the HP was operated in a given hour; *price* - price for electricity within a given hour. The temperature dynamics of the house is described by the following discrete linear time-invariant state space model:

$$intemp(t+1) = (1 - rc1) \cdot intemp(t) + rc1 \cdot outtemp + nc \cdot hppower \quad (2)$$

where  $rc1$  and  $nc$  are model parameters, the values of which are specific this specific house.

**Task 7.1** Use the historical data and the SolveDB *shared model* for the HP stored in the database table *models* for fine-tuning the HP model parameters  $rc1$  and  $nc$ .

**Task 7.2** Defined and solve an optimization problem to minimize the HP operating costs